

Dell™ Remote Access Controller

# Racadm User's Guide



## Notes, Notices, and Cautions



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



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



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

---

**Information in this document is subject to change without notice.**

**© 2002–2003 Dell Inc. All rights reserved.**

Reproduction in any manner whatsoever without the written permission of Dell Inc. is strictly forbidden.

Trademarks used in this text: *Dell*, the *DELL* logo, and *PowerEdge* are trademarks of Dell Inc; *Microsoft*, *Windows*, and *MS-DOS* are registered trademarks of Microsoft Corporation; *Novell* and *NetWare* are registered trademarks of Novell Corporation; *Red Hat* is a registered trademark of Red Hat, Inc.; *UNIX* is a registered trademark of The Open Group in the United States and other countries; *Intel* and *Itanium* are registered trademarks of Intel Corporation.

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 Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

# Contents

## 1 Serial and Racadm Commands

<b>Serial/Telnet Commands</b> . . . . .	<b>9</b>
<b>racadm Command</b> . . . . .	<b>11</b>
racadm . . . . .	11
Description . . . . .	11
Before Using the Racadm Remote Capability . . . . .	11
Using racadm Remotely . . . . .	11
racadm Remote Security . . . . .	12
racadm Synopsis . . . . .	12
racadm Options . . . . .	12
racadm Subcommands . . . . .	13
racadm Error Messages . . . . .	14
<b>racadm Subcommand Man Pages</b> . . . . .	<b>15</b>
help . . . . .	15
allowEnDisRem . . . . .	16
clearasrscreen . . . . .	16
config/getconfig . . . . .	17
coredump . . . . .	20
coredumpdelete . . . . .	21
fwupdate . . . . .	22
getsensors . . . . .	25
getssninfo . . . . .	30
getsysinfo . . . . .	32
setractime/getractime . . . . .	34
setniccfg/getniccfg . . . . .	35
getsctag . . . . .	36
setassettag/getassettag . . . . .	37
localauthenable . . . . .	37
racdump . . . . .	38
racreset . . . . .	39
racresetcfg . . . . .	40
serveraction . . . . .	41
getmodinfo . . . . .	42
getraclog . . . . .	43

clrraclog . . . . .	44
getsel . . . . .	44
clrsel . . . . .	45
connect . . . . .	45
gettracelog . . . . .	47
setrac . . . . .	48
sslcsrgen . . . . .	49
sslcertupload . . . . .	50
sslcertview . . . . .	51
syncvncpassword . . . . .	53
<b>racadm Test Subcommands . . . . .</b>	<b>53</b>
testalphapage . . . . .	53
testemail . . . . .	54
testnumpage . . . . .	55
testtrap . . . . .	56
<b>Using the racadm Utility to Configure the RAC . . . . .</b>	<b>56</b>
Locating the Proper User Index . . . . .	57
Adding the RAC User Without Alert Capabilities. . . . .	57
Deleting a RAC User. . . . .	58
Adding a RAC User With Alerting Capabilities. . . . .	58
Configuring RAC Network Properties. . . . .	60
Locating the Proper Dial-In (PPP) User Index. . . . .	61
Adding a Dial-In (PPP) User . . . . .	61
Locating the Proper Demand Dial-Out Entry Index. . . . .	62
Adding a Demand Dial-Out Entry . . . . .	63
Configuring Dial-In and Dial-Out (PPP) Properties. . . . .	64
<b>Property Database . . . . .</b>	<b>65</b>
Database Properties . . . . .	66
Configuration Using the racadm Utility and the racadm.cfg File . . . . .	74
<b>RAC Configuration File . . . . .</b>	<b>76</b>
Parsing Rules . . . . .	77

## 2 RAC Property Database Group and Object Definitions

<b>idRacInfo . . . . .</b>	<b>81</b>
idRacType (Read Only). . . . .	81
idRacProductInfo (Read Only) . . . . .	81

idRacDescriptionInfo (Read Only) . . . . .	82
idRacVersionInfo (Read Only) . . . . .	82
idRacName (Read/Write) . . . . .	82
idPcmcialInfo (Read Only) . . . . .	83
idRacMisc (Read/Write) . . . . .	83
<b>cfgMonitor . . . . .</b>	<b>83</b>
cfgMonitorExtPwrScrRule (Read/Write) . . . . .	83
cfgMonitorBatteryRule (Read/Write) . . . . .	84
<b>cfgLanNetworking . . . . .</b>	<b>84</b>
cfgNicEnable (Read/Write) . . . . .	84
cfgNicIpAddress (Read/Write) . . . . .	84
cfgNicNetmask (Read/Write) . . . . .	85
cfgNicGateway (Read/Write) . . . . .	85
cfgNicUseDhcp (Read/Write) . . . . .	85
<b>cfgCurrentLanNetworking . . . . .</b>	<b>86</b>
cfgNicCurrentIpAddress (Read Only) . . . . .	86
cfgNicCurrentNetmask (Read Only) . . . . .	86
cfgNicCurrentGateway (Read Only) . . . . .	86
cfgNicCurrentDhcpWasUsed (Read Only) . . . . .	87
<b>cfgDialupNetworking . . . . .</b>	<b>87</b>
cfgDialupDialInEnable (Read/Write) . . . . .	87
cfgDialupDialOutEnable (Read/Write) . . . . .	87
cfgDialupDialInDhcpEnable (Read/Write) . . . . .	88
cfgDialupDialInBaseIpAddr (Read/Write) . . . . .	88
cfgDialupDialInDfltIpAddr (Read/Write) . . . . .	88
cfgDialupDialIdleTimeout (Read/Write) . . . . .	89
cfgDialupDiMaxConnectTime (Read/Write) . . . . .	89
cfgDialupDialType (Read/Write) . . . . .	89
cfgDialupModemInitString (Read/Write) . . . . .	89
cfgDialupModemBaudRate (Read/Write) . . . . .	90
cfgDialupDialInAuthType (Read/Write) . . . . .	90
cfgDialupDialOutModemConnectTimeout (Read/Write) . . . . .	90
<b>cfgRemoteHosts . . . . .</b>	<b>90</b>
cfgRhostsSntpEmailEnable (Read/Write) . . . . .	91
cfgRhostsFloppyTftpEnable (Read/Write) . . . . .	91
cfgRhostsFwUpdateTftpEnable (Read/Write) . . . . .	91
cfgRhostsSntpServerIpAddr (Read/Write) . . . . .	91

cfgRhostsFloppyTftpIpAddr (Read/Write) . . . . .	92
cfgRhostsFloppyTftpPath (Read/Write) . . . . .	92
cfgRhostsFwUpdateIpAddr (Read/Write) . . . . .	92
cfgRhostsFwUpdatePath (Read/Write) . . . . .	93
<b>cfgUserAdmin . . . . .</b>	<b>93</b>
cfgUserAdminUserName (Read/Write) . . . . .	93
cfgUserAdminPassword (Write Only) . . . . .	94
cfgUserAdminAlertFilterRacEventMask (Read/Write) . . . . .	94
cfgUserAdminAlertFilterSysEventMask (Read/Write) . . . . .	94
cfgUserAdminPageNumericEnable (Read/Write) . . . . .	94
cfgUserAdminPageNumericNumber (Read/Write) . . . . .	95
cfgUserAdminPageNumericMessage (Read/Write) . . . . .	95
cfgUserAdminPageNumericHangupDelay (Read/Write) . . . . .	95
cfgUserAdminPageAlphaEnable (Read/Write) . . . . .	96
cfgUserAdminPageAlphaNumber (Read/Write) . . . . .	96
cfgUserAdminPageAlphaProtocol (Read/Write) . . . . .	96
cfgUserAdminPageAlphaBaudRate (Read/Write) . . . . .	97
cfgUserAdminPageAlphaCustomMsg (Read/Write) . . . . .	97
cfgUserAdminPageAlphaModemConnectTimeout (Read/Write) . . . . .	97
cfgUserAdminPageAlphaPagerId (Read/Write) . . . . .	98
cfgUserAdminPageAlphaPassword (Read/Write) . . . . .	98
cfgUserAdminEmailEnable (Read/Write) . . . . .	98
cfgUserAdminEmailAddress (Read/Write) . . . . .	99
cfgUserAdminEmailCustomMsg (Read/Write) . . . . .	99
cfgUserAdminPageModemInitString (Read/Write) . . . . .	99
<b>cfgTraps . . . . .</b>	<b>99</b>
cfgTrapsDestIpAddr (Read/Write) . . . . .	100
cfgTrapsEnable (Read/Write) . . . . .	100
cfgTrapsSnmpCommunity (Read/Write) . . . . .	100
cfgTrapsFilterRacEventMask (Read/Write) . . . . .	100
cfgTrapsFilterSysEventMask (Read/Write) . . . . .	101
<b>cfgDemandDial . . . . .</b>	<b>101</b>
cfgDemandDialIpAddr (Read/Write) . . . . .	101
cfgDemandDialPhoneNumber (Read/Write) . . . . .	102
cfgDemandDialPppUserName (Read/Write) . . . . .	102
cfgDemandDialPppPassword (Read/Write) . . . . .	102
cfgAuthType (Read/Write) . . . . .	102
<b>cfgDialinUsers . . . . .</b>	<b>103</b>


cfgDialinUsrPppUserName (Read/Write) . . . . .	103
cfgDialinUsrPppUserPassword (Read/Write) . . . . .	103
cfgDialinUsrCallbackType (Read/Write) . . . . .	104
cfgDialinUsrCallbackNumber (Read/Write) . . . . .	104
<b>cfgSessionManagement . . . . .</b>	<b>104</b>
cfgSsnMgtMaxConRedirSessions (Read/Write) . . . . .	104
cfgSsnMgtMaxSessions (Read/Write) . . . . .	105
cfgSsnMgtMaxSessionsPerUser (Read/Write) . . . . .	105
<b>cfgSerial . . . . .</b>	<b>105</b>
cfgSerialBaudRate (Read/Write) . . . . .	105
cfgSerialConsoleEnable (Read/Write) . . . . .	106
cfgSerialConsoleQuitKey (Read/Write) . . . . .	106
cfgSerialConsoleIdleTimeout (Read/Write) . . . . .	106
cfgSerialConsoleShellType (Read/Write) . . . . .	107
cfgSerialConsoleNoAuth (Read/Write) . . . . .	107
cfgSerialConsoleCommand (Read/Write) . . . . .	107
cfgSerialTelnetEnable (Read/Write) . . . . .	108
cfgSerialConsoleColor (Read/Write) . . . . .	108
cfgSerialConsoleAnsi (Read/Write) . . . . .	108
cfgSerialTelnetColor (Read/Write) . . . . .	108
cfgSerialTelnetAnsi (Read/Write) . . . . .	109
cfgSerialCom2RedirEnable (Read/Write) . . . . .	109
<b>cfgNetTuning . . . . .</b>	<b>109</b>
cfgNetTuningNicMtu (Read/Write) . . . . .	109
cfgNetTuningIptTtl (Read/Write) . . . . .	110
cfgNetTuningIpreassTtl (Read/Write) . . . . .	110
cfgNetTuningTcpSrttBase (Read/Write) . . . . .	110
cfgNetTuningTcpSrttDflt (Read/Write) . . . . .	111
cfgNetTuningTcpReXmtMin (Read/Write) . . . . .	111
cfgNetTuningTcpReXmtMax (Read/Write) . . . . .	111
cfgNetTuningTcpMsl (Read/Write) . . . . .	111
Tuning the RAC for Satellite Connectivity . . . . .	112
<b>cfgOobSnmp . . . . .</b>	<b>113</b>
cfgOobSnmpTrapsEnable (Read/Write) . . . . .	113
cfgOobSnmpAgentEnable (Read/Write) . . . . .	113
<b>cfgRacTuning . . . . .</b>	<b>113</b>
cfgRacTuneMnNwIpaAddrBase (Read/Write) . . . . .	114

cfgRacTuneHostVncPort (Read/Write) . . . . .	114
cfgRacTuneGenericFtpUser (Read/Write) . . . . .	114
cfgRacTuneGenericFtpPasswd (Read/Write) . . . . .	114
cfgRacTuneRemoteCrsPort (Read/Write) . . . . .	115
cfgRacTuneFwUpdateResetDelay (Read/Write) . . . . .	115
cfgRacTuneDialOutIdleTime (Read/Write) . . . . .	115
cfgRacTuneDialOutMaxConnectTime (Read/Write) . . . . .	116
cfgRacTuneD3debugEnable (Read/Write) . . . . .	116
cfgRacTuneRemoteRacadmEnable (Read/Write) . . . . .	116
cfgRacTuneMnConsoleDisableAllowed . . . . .	117
cfgRacTuneHostCom2BaudRate (value) . . . . .	117
<b>ifcRacManagedNodeOs . . . . .</b>	<b>117</b>
ifcRacMnOsHostname (Read/Write) . . . . .	117
ifcRacMnOsOsName (Read/Write) . . . . .	118
ifcRacMnOsOsType (Read Only) . . . . .	118
<b>cfgRacSecurity . . . . .</b>	<b>118</b>
cfgRacSecCapabilities (Read Only) . . . . .	118
cfgRacSecCsrKeySize (Read Only) . . . . .	119
cfgRacSecCsrCommonName (Read/Write) . . . . .	119
cfgRacSecCsrOrganizationName (Read/Write) . . . . .	119
cfgRacSecCsrOrganizationUnit (Read/Write) . . . . .	119
cfgRacSecCsrLocalityName (Read/Write) . . . . .	120
cfgRacSecCsrStateName (Read/Write) . . . . .	120
cfgRacSecCsrCountryCode (Read/Write) . . . . .	120
cfgRacSecCsrEmailAddr (Read/Write) . . . . .	120
cfgRacSecVncInEncryptEnable (Read/Write) . . . . .	121
cfgRacSecAuthLocalRacEnable (Read Only) . . . . .	121
cfgRacSecAuthLocalOsEnable (Read/Write) . . . . .	121

Abbreviations and Acronyms . . . . .	123
--------------------------------------	-----

# Serial and Racadm Commands


Your remote access controller (RAC) provides a serial or telnet console and racadm command-line interface that you can use to enter commands to manage and configure your system locally or remotely.


 **NOTE:** For information about accessing and using the RAC serial/telnet console, see your *RAC User's Guide*.

The serial/telnet console provides a set of serial commands. The serial commands, which include the **racadm** command, give you access to all of the features supported by the RAC Web-based interface. Therefore, you have full access to all of the features of the RAC and management of the server through the serial or telnet console.

The racadm command-line utility provides a scriptable interface that allows you to locally or remotely configure your RAC. The racadm utility runs on the management station and the managed system and is available on the *Dell OpenManage Systems Management CD* (32-bit systems) and on the *Dell OpenManage Server Management CD* (64-bit systems).

You can also use the racadm utility to write scripts to automatically configure multiple RACs. For example, when configuring in batch mode, a series of calls to the racadm utility can perform actions such as upgrading RAC firmware, setting a property to a specific value, and reading the configuration of the RAC. All command switches are *not* case-sensitive. For more information about configuring multiple RACs, see "RAC Configuration File."

 **NOTE:** The racadm command-line utility is supported only on Dell™ PowerEdge™ 7150 systems running Microsoft Windows Advanced Server Limited Edition. However, when you use the utility on this system, the primary command is **drscadm** instead of **racadm**. For more information about using the drscadm utility on PowerEdge 7150 systems, see the *Dell Remote Service Card User's Guide* on your documentation CD.

 **NOTE:** You can also use the Option ROM on your RAC or Dell OpenManage™ Server Administrator to configure your RAC.

For more information about your RAC, including a list of other documents you may need, see your *Dell Remote Access Controller Installation and Setup Guide*.

## Serial/Telnet Commands

The serial commands in Table 1-1 can be run from the serial or telnet console command prompt or the racadm CLI utility locally or remotely. To access the serial/telnet console command prompt, see your *Dell Remote Access Controller Installation and Setup Guide*.

Enter the **help** command to display the entire serial/telnet command list. Commands that are not supported on the system or interface that you are using are labeled as such. For example, if the **getsensors** command is not supported on the system, the following text is displayed next to the command:

```
<NOT SUPPORTED>
```

If you try to enter a command that is not supported on the system you are using, an error similar to the following is displayed.

```
setsvctag: Firmware: UNSUPPORTED COMMAND
```

Table 1-1 lists the serial/telnet commands. These commands are also supported as **racadm** subcommands. The descriptions and "man page" information including required syntax for the serial telnet commands are identical for the **racadm** command. You do not need to type **racadm** before entering a serial/telnet command because the serial/telnet commands are not **racadm** subcommands. They are at the same "level." For detailed information about the required syntax for each **racadm** subcommand see "racadm Subcommand Man Pages."


**Table 1-1. Serial/Telnet Commands**


<b>Command</b>	<b>Description</b>
<b>help</b>	Lists RAC subcommands.
<b>help &lt;subcommand&gt;</b>	Lists usage statement for the specified subcommand.
<b>logout</b>	Logs out of a RAC session and then prints a new log in prompt.
<b>quit</b>	Logs out of a RAC session and then prints a new log in prompt.
<b>exit</b>	Logs out of a RAC session and then prints a new log in prompt.
<b>getsysinfo</b>	Displays general RAC and system information.
<b>getsensors</b>	Displays RAC sensor readings and information.
<b>serveraction</b>	Executes a graceful hard server reset, or power-on, power-off, or power-cycle.
<b>getmodinfo</b>	Gets module configuration and status information.
<b>getraclog</b>	Displays RAC log entries.
<b>clr raclog</b>	Completely clears the RAC log entries.
<b>gettracelog</b>	Displays Trace Log entries.
<b>getsel</b>	Displays System Event Log entries.
<b>clrsel</b>	Completely clears the System Event Log entries.
<b>connect</b>	Connects the RAC to the COM2 serial port or video text console
<b>racadm</b>	Command-line status and configuration utility for RACs.


## racadm Command

Use the **racadm** command to enter subcommands to configure RAC properties. When you enter **racadm** subcommands (see Table 1-2), the **racadm** utility sets or retrieves object property values.

For the **racadm** interface, entering the **racadm help** command displays the entire **racadm** subcommand command list, which lists all the subcommands supported by any RAC product.

 **NOTE:** MS-DOS® version 6.22 is required to use the DOS **racadm** command-line feature. To limit DOS **racadm** output to a single screen, use the MS-DOS **more.com** utility. Type the following command at the prompt: `a: racadm help | more.`

 **NOTE:** To limit the processing load on the managed system and the RAC when using the **racadm** utility, add one- or two-second **sleep** or **delay** commands between each **racadm** command.

 **NOTICE:** Commands and code samples in this section are RAC-specific. Some commands or samples may not be applicable to your RAC. For example, lines that contain configuration code for modems, dial-in users, or demand dial-out entries are specific to DRAC III.

### racadm

**Table 1-2. racadm command**

Command	Definition
racadm	Command-line status and configuration utility for RACs.

### Description

Without options, **racadm** executes the **help** command, which displays a list of available tests and a one-line description of each. The **racadm help <subcommand>** displays any syntax and command-line options.

### Before Using the Racadm Remote Capability

Configure the IP address on your RAC before using the **racadm** remote capability. For more information about initially configuring your RAC, including a list of other documents you may need, see your *Dell Remote Access Controller Installation and Setup Guide*.

### Using racadm Remotely

The **racadm** utility provides a remote capability option (**-r**) that allows you to connect to the managed system and execute **racadm** subcommands from a remote console or management station. To use the remote capability, you need a valid user name (**-u** option) and password (**-p** option), and the IP address of the managed system.

 **NOTE:** The **racadm** remote capability is supported only on management stations running Microsoft® Windows® 2000, Windows Server 2003, and Red Hat® Linux operating systems.

## racadm Remote Security

The racadm remote interface uses CHAP MD5 encrypted authentication.

### racadm Synopsis

```
racadm <options> <subcommand> <subcommand_options>
```



**NOTICE:** The user name must be **root** or **racadmuser** (for authentication) when using this command. Any other user receives an authentication error.

```
racadm <options> [-u <user name>] -p <password> -r <racIpAddr>
<subcommand>
```

```
racadm <options> -i -r <racIpAddr> <subcommand>
```

```
racadm <options> -r <racIpAddr> <subcommand>
```

### racadm Options

The base racadm command has the options shown in Table 1-3.

**Table 1-3. racadm Command Options**

Option	Description
-r <racIpAddr>	Specifies the remote IP address of the controller.
-i	Tells racadm to interactively query the user for the user's user name and password.
-u <usrName>	Specifies the user name that is used to authenticate the command transaction. If not specified, the default user name "racadmusr" is used. If the -u option is used, the -p option must be used, and the -i option (interactive) is not allowed.
-p <password>	Specifies the password used to authenticate the command transaction. If the -p option is used, the -i option is not allowed.
-l <lvl>	Specifies the log level for debug. <b>NOTE:</b> If you use the -l <lvl> option without using the -L <file> option, a default logfile named <b>drscadm.log</b> is created in the current working directory.
-v <lvl>	Specifies the verbose level for screen output.
-t <secs>	Specifies the transport time-out.
-L <file>	Specifies a debug log file.

If you use the -r option, you must also use the -u and -p options to configure the RAC to accept racadm commands. Using the -r option without the previously listed options will result in a command failure.

## Enabling and Disabling Remote Security



**NOTE:** It is recommended that you run these commands on your local system. You must reset the RAC controller to enable these commands.

Enter the following command to enable remote security:

```
racadm config -g cfgRacTuning -o cfgRacTuneRemoteRacadmEnable 1
```

Enter the following command to disable remote security:

```
racadm config -g cfgRacTuning -o cfgRacTuneRemoteRacadmEnable 0
```

## Log/Verbose Levels

Use the log/verbose level options in Table 1-4 to control how **racadm** command output is displayed.

**Table 1-4. Log/Verbose Level Options**

Option	Description
0x1	Standard out messages
0x2	Standard error messages
0x4	Debug messages

Example:

```
racadm -l 0x3 -L log <subcommand [subcommand_options]>
```

The argument 0x3 for the log level is the OR of STDOUT and STDERR messages. Both of these message types are written to the file name **log**. The verbose, or **-v**, option defaults to 0x3, or STDOUT, STDERR.

## racadm Subcommands

The following subsections provide descriptions of subcommands that you can run under the **racadm** command-line utility. The primary reason that a command is limited as a **racadm** subcommand is to enforce that only "root" and "racadmuser" users can execute these commands. Table 1-5 briefly describes each **racadm** subcommand. For detailed information about each subcommand including syntax and valid entries, see "racadm Subcommand Man Pages."

**Table 1-5. racadm Subcommands**

Command	Description
help	Lists RAC subcommands.
help <subcommand>	Lists usage statement for the specified subcommand.
allowEnDisRem	Allows enabling or disabling of the remote console.
clearasrscreen	Clears the last ASR screen (last blue screen).

**Table 1-5. racadm Subcommands**

config/getconfig	Configures the RAC and displays the RAC configuration.
coredump	Displays the last RAC coredump.
coredumpdelete	Deletes the core dump stored in the RAC.
fwupdate	Executes or displays status on RAC firmware updates.
Command	Description
getsensors	Displays RAC sensor readings and information.
getssninfo	Displays information about active sessions.
getsysinfo	Displays general RAC and system information.
setractime/getractime	Sets or displays the RAC time.
setniccfg/getniccfg	Sets or displays the current RAC IP configuration.
getsvctag	Displays service tags.
setassettag/getassettag	Sets and displays asset tags.
localauthenable	Enables or disables local operating system authentication.
racdump	Dumps RAC status and state information for debug.
racreset	Resets the RAC.
racresetcfg	Resets the RAC to the default configuration.
serveraction	Executes a graceful hard server reset, or power-on, power-off, or power-cycle.
setrac	Sets time, managed system name, operating system name, and type from the managed system to the RAC.
sslcsngen	Generates and download the secure sockets layer (SSL) certificate signing request (CA).
sslcertupload	Uploads a CA certificate or server certificate to the RAC.
sslcertview	Views a CA certificate or server certificate in the RAC.
syncvncpassword	Synchronizes the VNC password from the RAC to the host VNC.
testalphapage	Sends an alphanumeric page to a user.
testemail	Forces the RAC to send an e-mail over the RAC network adapter.
testnumpage	Sends an alphanumeric page to a user.
testtrap	Forces the RAC to send an SNMP over the RAC network adapter.

### **racadm Error Messages**

When using **racadm** commands and subcommands, you may encounter one or more of the following errors:

- Local error messages – Occur when problems occur with syntax, typographical errors, incorrect names, and so on.

Example:

```
racadm <subcmd>: ERROR: <message>
```

- Transport error messages – Occur when the racadm communication paths to the RAC are not accessible. Transport error messages occur if Server Administrator is not running when the command is executed or the Remote Access Service (RAS) is not set up.

If you are running racadm remotely, transport error messages may indicate network communication problems or that the management station is unable to communicate with the RAC.

Example:

```
racadm <subcmd> : Transport : ERROR : <message>
```

- RAC firmware errors – Occur when a fault exists in the RAC firmware operation.

Example:

```
racadm <subcmd> : Firmware : ERROR : <message>
```

## racadm Subcommand Man Pages

This section provides descriptions of subcommands that you can run under the racadm command-line utility.

### help

Table 1-6 describes the **help** command.

**Table 1-6. Help Command**

Command	Definition
help	Lists all of the subcommands available to use with <b>racadm</b> and provides a short description for each.

### Synopsis

```
racadm help
```

```
racadm help <subcommand>
```

### Description

The **help** subcommand lists all of the subcommands that are available under the **racadm** command along with a one-line description. You may also enter a subcommand after **help** to get the syntax for a specific subcommand.

**Input**

None

**Output**

The `racadm help` command displays a complete list of subcommands.

The `racadm help <subcommand>` command displays usage information for the specified subcommand only.

**allowEnDisRem**

Table 1-7 describes the `allowEnDisRem` subcommand.

**Table 1-7. allowEnDisRem**

Subcommand	Definition
<code>allowEnDisRem</code>	Allows enabling or disabling of the remote console.

**Synopsis**

```
racadm allowEnDisRem <TRUE | FALSE>
```

**Description**

The `allowEnDisRem` subcommand is applicable only to Windows operating systems. When the `TRUE` message is displayed, a user performing console redirection through the RAC Web-based remote access interface can disable the local keyboard and mouse. When the `FALSE` message is displayed, the local keyboard and mouse cannot be disabled.

**Input**

None

**Output**

None

**clearasrscreen**

Table 1-8 describes the `clearasrscreen` subcommand.

**Table 1-8. clearasrscreen**

Subcommand	Definition
<code>clearasrscreen</code>	Clears the last crash screen that is in memory.

## Synopsis

racadm clearasrscreen

## Input

None

## Output

None

## config/getconfig

Table 1-9 describes the **config** and **getconfig** subcommands.

**Table 1-9. config/getconfig**

Subcommand	Definition
config	Configures the RAC.
getconfig	Gets the RAC configuration data.

## Synopsis

```
racadm config [-s -p -c] -f <filename>
```

```
racadm config [-s] -g <groupName> -o <objectName> [-i <index>]  
<Value>
```

```
racadm getconfig [-p] -f <filename>
```

```
racadm getconfig -g <groupName> [-i <index>]
```

```
racadm getconfig -u <username>
```

```
racadm getconfig -h
```

## config Subcommand Description

The **config** subcommand allows the user to set RAC configuration parameters individually or to batch them as part of a configuration file. After the **.cfg** file has been correctly parsed, each object is read. If the content is the same, a write to the RAC does not occur. If the data is different, that RAC object is written with the new value.

## Input

Table 1-10 describes the **config** subcommand options.



**NOTE:** The **-f**, **-s**, and **-p** options are not supported for the serial/telnet console.

**Table 1-10. config Subcommand Options and Descriptions**

Option	Description
-f	The <b>-f</b> <filename> option causes config to read the contents of the file specified by <filename> and configure the RAC. The file must contain data in the format specified in "Parsing Rules."
-s	The <b>-s</b> , or synchronize option, directs <b>config</b> to synchronize the user and password database with Server Administrator (if any user passwords were modified).
-p	The <b>-p</b> , or password option, directs <b>config</b> to delete the password entries contained in the config file <b>-f</b> <filename> after the configuration is complete.
-g	The <b>-g</b> <groupName>, or group option, must be used with the <b>-o</b> option. The <groupName> specifies the group containing the object that is to be set.
-o	The <b>-o</b> <objectName> <Value>, or object option, must be used with the <b>-g</b> option. This option specifies the object name that is written with the string <Value>.
-i	The <b>-i</b> <index>, or index option, is only valid for indexed groups and can be used to specify a unique group. The <index> is a decimal integer from 1 through 16. The index is specified here by the index value, not a "named" value.
-c	The <b>-c</b> , or check option, is used with the <b>config</b> subcommand and allows the user to parse the .cfg file to find syntax errors. If errors are found, the line number and a short description of what is incorrect are displayed. Writes do not occur to the RAC. This option is a check only.

### Output

This subcommand generates error output upon encountering either of the following:

- Invalid syntax, group name, object name, index, or other invalid database members
- racadm utility transport failures

If errors are not encountered, this subcommand returns an indication of how many configuration objects that were written out of how many total objects were in the .cfg file.

### getconfig Subcommand Description

The **getconfig** subcommand allows the user to retrieve RAC configuration parameters on an individual basis, or all the configuration groups may be retrieved and saved into a file.

### Input

Table 1-11 describes the **getconfig** subcommand options.

 **NOTE:** The **-f** option is not supported for the serial/telnet console.

**Table 1-11. getconfig Subcommand Options**

Option	Description
-f	The <b>-f &lt;filename&gt;</b> , or <b>filename</b> option, causes <b>getconfig</b> to create the file <b>&lt;filename&gt;</b> . It then reads all the configuration data from the RAC and places it into the file <b>&lt;filename&gt;</b> . The created file is a format that can be used with the <b>racadm config -f &lt;filename&gt;</b> command.
-p	The <b>-p</b> , or <b>password</b> option, causes <b>getconfig</b> to include password information in the file for all passwords except for the user passwords (which are stored encrypted and cannot be decrypted). A <b># cfgUserAdmPassword</b> line is displayed as an indication that the password is present, but the password text is not displayed.
-g	The <b>-g &lt;groupName&gt;</b> , or <b>group</b> option, can be used to display the configuration for a single group. The <b>groupName</b> is the name for the group used in the <b>racadm.cfg</b> files. If the group is an indexed group, use the <b>-i</b> option.
-h	The <b>-h</b> , or <b>help</b> option, displays a list of all available configuration groups so the user may enter the desired group. This option is useful when you do not remember exact group names.
-i	The <b>-i &lt;index&gt;</b> , or <b>index</b> option, is valid only for indexed groups and can be used to specify a unique group. The <b>&lt;index&gt;</b> is a decimal integer from 1 through 16. If <b>-i &lt;index&gt;</b> is not specified, a value of 1 is assumed for groups, which are tables that have multiple entries. The index is specified by the index value, not a "named" value.
-u	The <b>-u &lt;username&gt;</b> , or <b>user name</b> option, can be used to display the configuration for the specified user. The <b>&lt;username&gt;</b> option is the log in user name for the user.

## Output

This subcommand generates error output upon encountering either of the following:

- Invalid syntax, group name, object name, index, or other invalid database members
- racadm utility transport failures

If errors are not encountered, this subcommand displays the contents of the specified configuration.

## Examples

- **racadm getconfig -g cfgLanNetworking** – Displays all of the configuration parameters (objects) that are contained in the group **cfgLanNetworking**.
- **racadm config -g cfgLanNetworking -o cfgNicIpAddress 10.35.10.100** – Sets the **cfgNicIpAddress** configuration parameter (object) to the value 10.35.10.110. This IP address object is contained in the group **cfgLanNetworking**.
- **racadm getconfig -f myrac.cfg** – Reads *all* of the configuration objects, from all of the RAC group configuration parameters and place them in a file called **myrac.cfg**.

- `racadm config -f myrac.cfg` – Configures or reconfigures the RAC. The `myrac.cfg` file may be created from the command specified in the previous example if the RAC has a desired configuration to be replicated. The `myrac.cfg` file may also be edited manually as long as the parsing rules are followed.



**NOTE:** The `myrac.cfg` file does not contain password information. To include this information in the file, it must be input manually. If you want to remove password information from the `myrac.cfg` file, use the `-p` option.

- `racadm getconfig -h` – Displays a list of the available configuration groups on the RAC.
- `racadm getconfig -u root` – Displays the configuration parameters for the user named root.

## coredump

Table 1-12 describes the `coredump` subcommand.

**Table 1-12. coredump**

Subcommand	Definition
<code>coredump</code>	Displays the last RAC core dump.

### Synopsis

```
racadm coredump
```

### Description

The `coredump` subcommand displays detail information, including register values, recorded when the most recent bus error occurred, or the message `No CORE dump available` (if a previous bus error has not occurred, or if the data has been cleared).

This bus error information is persistent across power cycles of the RAC—the data remains in the flash memory of the RAC until either:

- It is cleared using the `coredumpdelete` subcommand.
- Another bus error occurs, replacing the previous information with the more recent bus error information.

See the `coredumpdelete` subcommand for information about deleting this information.

### Input

None

### Output

Example:

```
FW d_cmdCoreDump:
```

```
Last CORE dump at Wed, 23 Oct 2002 15:49:41 GMT-05:00
```

Data Access Abort Running: 'IPEV' -#005E0000

-----  
CPSR = A0000013 (NzCv if SVC32) SP =018616DC LR =01023C34  
R0 =D000AEB2 R1 =01151C1C R2 =0186179C R3 =00000007 R4 =01861700  
R5 =C854E35C R6 =018617A0 R7 =00000011 R8 =01383C44 R9 =F1C729C6  
R10=00000004 R11=01151C1C R12=0000005F USP=DEADBEEF ULR=00000000  
SSP=018616DC SLR=01023C34 SPSR\_svc=60000013  
PC =01151C88-01151C88:

Image ID: jon Wed Oct 23 14:55:53 CDT 2002  
//DG0YN811/c/esm4/10\_23/RAC2.0/FW/rmc

Sysuptime: 67f

FIQ stack

<binary data>

IRQ stack

<binary data>

UNDEF stack

<binary data>

ABORT stack

<binary data>

SVC stack

<binary data>

SWI stack

<binary data>

Ienter stack

<binary data>

Current task stack: 'IPEV' -#005e0000

<binary data>

where <binary data> is the binary information that was generated by the RAC firmware.

Example output (when data is not available):

FW d\_cmdCoreDump: No CORE dump available.

## coredumpdelete

Table 1-13 describes the **coredumpdelete** subcommand.

**Table 1-13. coredumpdelete**

Subcommand	Definition
coredumpdelete	Deletes the core dump stored in the RAC.

**Synopsis**

```
racadm coredumpdelete
```

**Description**

The `coredumpdelete` command clears the area reserved for persistent storage of bus error information. This operation is performed regardless of whether any bus error information is currently stored in the area reserved for this information.

See the `coredump` command for information about displaying the bus error information.

**Input**

None

**Output**

None

**fwupdate**

Table 1-14 describes the `fwupdate` subcommand.

**Table 1-14. fwupdate**

Subcommand	Definition
fwupdate	Updates the firmware on the RAC.

**Using Firmware Version 1.0 or 1.1**

If you are using firmware version 1.0 or 1.1 for the DRAC III controller only, use the `-d` option to upgrade your firmware.

If you use the `-f` option, specify the proper filename for the installed RAC. These filenames include:

- `firmimg.bin` (DRAC III)
- `firmimg.rm1` (ERA/O)
- `firmimg.bm1` (ERA and DRAC III/XT)

If you are updating your firmware *locally*, use one of the following commands to update your firmware:

```
racadm fwupdate -p -u -d <directory>
```

```
racadm fwupdate -p -u -f <filename>
```

**Example:**

```
racadm fwupdate -p -u -d \my\updatefiles\path  
racadm fwupdate -p -u -f \my\updatefiles\path\<filename>
```

If you are updating your firmware *remotely*, use the following command to update your firmware:

```
racadm -r <RAC_IP> -u <user> -p <password> fwupdate -g -u -a  
<TFTP_IP> -d <TFTP_dir_path>
```



**NOTE:** The `-p` option does not support remote firmware updates.

### Using Firmware Version 1.2 or Later

If you are using firmware version 1.2 or later for the DRAC III only, or if you are using any firmware level for the DRAC III/XT, ERA, or ERA/O, use the following command to upgrade your firmware:

```
racadm fwupdate -p -u -d <directory path>
```

The `racadm` command finds the appropriate firmware update file.

**Example:**

```
racadm fwupdate -p -u -d \my\updatefiles\path\
```

### Synopsis

```
racadm fwupdate -u [-w] [-D]
```

```
racadm fwupdate -s
```

```
racadm fwupdate -g [-u ] [-w][-D] [-a <IP address>]  
[-f <path/file>]
```

```
racadm fwupdate -c
```

```
racadm fwupdate -p -f <update filename> [-u] [-w] [-D]
```

### Description

The `fwupdate` subcommand allows the caller to update the firmware on the RAC. The user may:


- Start updating a firmware update file that has previously been loaded into the RAMdisk update area.
- Check the status of the firmware update process.
- Instruct the RAC firmware to get the firmware update file from a TFTP server and load it into the RAMdisk area.

The user may specify the IP address and path/filename or IP address and directory, or use the default values found in the database. The user may also specify that the update be started after loading, or to terminate and make a separate call to start the update process.

- Load the update file into the RAC RAMdisk area.

## Input

Table 1-15 describes the `fwupdate` subcommand options.

 **NOTE:** The `-p` and `-u` options are not supported for the serial/telnet console.

**Table 1-15. fwupdate Subcommand Options**

Option	Description
<code>-u</code>	The <code>update</code> option performs a checksum of the firmware update file and starts the actual update process. If this option is entered by itself, it is assumed that a firmware update file has already been loaded into the RAMdisk using the <code>-g</code> or <code>-p</code> options. This option may also be used along with the <code>-g</code> or <code>-p</code> options. After the firmware update file has been loaded, the update process is started within the same call. At the end of the update, the RAC performs a soft reset.
<code>-w</code>	The <code>wait</code> option represents a delay in seconds to wait before proceeding with the update. The <code>-w</code> option is only valid with the <code>-u</code> option.
<code>-s</code>	The <code>status</code> option returns the current status of where you are in the update process. This option is always entered by itself. Do not mix the <code>-s</code> options with others. If you do, the only action is the return of the status as if it was the only option entered.
<code>-g</code>	The <code>get</code> option instructs the firmware to get the firmware update file from the TFTP server and place it in the RAMdisk area. The user may also specify the <code>-a</code> and/or the <code>-f</code> or <code>-d</code> options that are described next. In the absence of the <code>-a</code> or <code>-f</code> options the defaults are read from properties contained in the group <code>cfgRemoteHosts</code> , using properties <code>cfgRhostsFwUpdateIpAddr</code> and <code>cfgRhostsFwUpdatePath</code> .
<code>-a</code>	The <code>IP Address</code> option specifies the IP address of the TFTP server.
<code>-f</code>	The <code>&lt;path/filename&gt;</code> option specifies the location on the TFTP server or on the RAC's host system where the firmware update file resides. Do not use the <code>-f</code> option with the <code>-d</code> option.
<code>-d</code>	The <code>-d</code> , or <code>directory</code> , option specifies the directory on the TFTP server or on the RAC's host server where the firmware update file resides. Do not use the <code>-f</code> option with the <code>-d</code> option.
<code>-c</code>	The <code>-c</code> , or <code>checksum</code> , option allows the user to verify an update file that has been loaded into the RAMdisk area. The update file can be loaded by one of the two loading mechanisms ( <code>racadm</code> utility or TFTP). This option essentially gets the size of the firmware update file and calculates the checksum, and verifies the file token. The TFTP interface verifies the checksum after loading automatically. This option is used mainly when using FTP. The <code>-c</code> option is not used with other options. (The <code>-u</code> option will <i>always</i> checksum before programming. It can be used along with the <code>-u</code> option).
<code>-D</code>	After the update is complete, the RAC is reset. Upon boot, a call is made to reset all firmware configuration parameters to the default values. For more information, see "racresetcfg."

**Table 1-15. fwupdate Subcommand Options**

Option	Description
-p	The <b>-p</b> , or <b>put</b> , option is used when you want to FTP the firmware update file from the managed system to the RAC. If the <b>-f</b> option is used, the name of the update image must be <b>firmimg.bin</b> . The update file is sent by way of FTP into the RAC. Checksum runs on the newly loaded image. If the checksum is not correct, an error message is displayed. The user is not required to enter the <b>fwupdate -s</b> option to do this. If the <b>-u</b> option is entered on the same command line, the update process starts.

### Output

Displays a message indicating which operation is being performed.

### Examples

- **racadm fwupdate -g -a 143.166.154.143 -f firmimg.bin**

In this example, the **-g** option tells the firmware to download the firmware update file from a location (specified by the **-f** option) on the TFTP server at a specific IP address (specified by the **-a** option). The update file is then loaded into RAMdisk. Since the **-u** option is not present, an update does *not* occur.

- **racadm fwupdate -s**

This option reads the current status of firmware update.

- **racadm fwupdate -u**

The **-u** option starts the update process. This command assumes that a valid firmware update file has been previously loaded using the **-g** or **-p** option. The update file checksum is verified for correctness before proceeding.

- **racadm fwupdate -g -u -a 143.166.154.143 -f firmimg.bin**

In this example, the **-g** option tells the firmware to download the firmware update file from a location (specified by the **-f** option) on the TFTP server at a specific IP address (specified by the **-a** option). The update file is then loaded into RAMdisk. The **-u** option tells the firmware to proceed with the update after the firmware is loaded.

### getsensors

Table 1-16 describes the **getsensors** subcommand.

**Table 1-16. getsensors**

Subcommand	Definition
getsensors	Gets information on the RAC sensors.

**Synopsis**

```
racadm getsensors
```

```
racadm getsensors -s <sensorNum>
```

```
racadm getsensors [-A] [-E] [-s <sensorNum>] [-h]
```

**Description**

The `getsensors` command displays information about the sensors that are present on the RAC itself, not in the system or on the BMC or ESM.

**Input**

Table 1-17 describes the `getsensors` subcommand options.

**Table 1-17. getsensors Subcommand Options**

Option	Description
-s	The <code>sensor</code> option allows the user to specify a single sensor to display. If an <code>-s</code> options is not used, all RAC sensors are displayed, one per line.
-A	The <code>-A</code> option generates output in API mode. No headers, hex output without leading 0x. The values <code>&lt;units&gt;</code> , <code>&lt;srcType&gt;</code> , <code>&lt;SrcStatus&gt;</code> , and <code>&lt;srcState&gt;</code> are output as hex numbers. The reading and other values as displayed as raw values.
-E	The <code>expanded output</code> option is only valid with the <code>-A</code> option, it cannot be used by itself.
-h	The <code>header</code> option displays text prior to dumping the sensor information. A large amount of information is included on the output line. The header text is cryptic due to the dense nature, but quite valuable after it is understood.

**Output**

Table 1-18 describes the `getsensors` subcommand options and expected output.

**Table 1-18. getsensors Subcommand Actions and Output**

Option	Output
Without the <code>-A</code> option	<pre>getsensors [-s &lt;sensorNum&gt;]</pre> <p>This command outputs the following on one line:</p> <pre>&lt;sensorNum&gt; &lt;sensorType&gt; &lt;sensorName&gt; &lt;reading&gt; &lt;units&gt; &lt;status&gt; &lt;srcType&gt; &lt;srcStatus&gt; &lt;srcState&gt;</pre> <p>where <code>&lt;units&gt;</code> and <code>&lt;sensorType&gt;</code> are expanded as strings, and the reading is given as a converted value (in volts or degrees Celcius).</p>

**Table 1-18. getsensors Subcommand Actions and Output**

Option	Output
With the -A option	<pre>getsensors -A -s 1</pre> <p>This command outputs the following on one line:</p> <pre>&lt;sensorNum&gt; &lt;sensorType&gt; &lt;sensorName&gt; &lt;reading&gt; &lt;units&gt; &lt;status&gt; &lt;srcType&gt; &lt;srcStatus&gt; &lt;srcState&gt;</pre> <p>No headers, hexadecimal output without leading 0x.</p>
With the -E option	<p>The -A output, plus the following, on one line:</p> <pre>&lt;norm&gt; &lt;min&gt; &lt;max&gt; &lt;lcThresh&gt; &lt;lncThresh&gt; &lt;uncThresh&gt; &lt;ucThresh&gt; &lt;m&gt; &lt;b&gt; &lt;K&gt; &lt;R&gt; &lt;posHyst&gt; &lt;negHyst&gt;</pre>

**General Information**

The output translation follows the formula:

$$y = 10^{**}R(m * X + B * 10^{**}K)$$

The text expansions and enumeration definitions are displayed in Table 1-19. The RAC sensors are implemented with the corresponding data.

**Table 1-19. RAC Sensor Number, Name, and Type**

<sensorNum>	<sensorType>	<sensorName>	<srcType>
1	Temperature	TEMP	"Temp"
2	Voltage	BATT VOLT	"Battery"
3	Voltage	EXTPWR VOLT	"Ext Pwr"
4	Voltage	+12PCI VOLT	"PCI Pwr"
5	Voltage	-12PCI VOLT	"PCI Pwr"
6	Voltage	+5PCI VOLT	"PCI Pwr"
7	Voltage	AUXPCI VOLT	"PCI Aux"

- <reading> – The 8-bit unsigned raw value. This value is converted to an actual temperature or voltage with the formula:  $y = 10^{**}R(m * X + B * 10^{**}K)$
- <units> – Indicates the units.
  - Degrees-Celcius, 1 (used for all temperature sensors)
  - Volts, 4 (used for all voltage sensors)
- <sensorType> – Enumeration values.
  - Temperature, 1

- Voltage, 2
- *<status>* – Enumeration values that indicate which region the sensor is in based on the thresholds.
  - Unknown, 0 (state at startup before sensors are read)
  - Normal, 1
  - Low-warning, 2 (low-noncritical)
  - High-warning, 3 (high-noncritical)
  - Low-critical, 4
  - High-critical, 5
  - Disabled, 6

The following are 8-bit unsigned raw reading values:

- *<norm>* – The normal value of the sensor
- *<min>* – The minimum value of the sensor
- *<max>* – The maximum value of the sensor
- *<lcThresh>* – The lower critical threshold
- *<lncThresh>* – The lower noncritical (warning) threshold
- *<uncThresh>* – The upper noncritical (warning) threshold
- *<ucThresh>* – The upper critical threshold

The following constants apply to the conversion formula:

- $y = 10^{**R}( m * X + B * 10^{**K} )$
- *<m>* – Slope: 10-bit 2's complement signed
- *<B>* – Intercept: 10-bit 2's complement signed
- *<K>* – B-exponent: 4-bit 2's complement signed
- *<R>* – R-exponent (result exponent): 4-bit 2's complement signed
- *<posHyst>* – Positive hysteresis value
- *<negHyst>* – Negative hysteresis value

The *<srcStatus>* and *<srcState>* values are interpreted based on the value of *<srcType>*. They indicate the status and state of the source that the sensor is monitoring. Table 1-20 shows the enumeration and valid values for *<srcStatus>* and *<srcState>* as a function of *<srcType>*.

**Table 1-20. Valid Values for "srcStatus" and "srcState" that Correspond to "srcType"**

<i>srcType</i>	<i>srcStatus</i>	<i>srcState</i>
"Temp" = 0	"" = 0	"" = 0

**Table 1-20. Valid Values for "srcStatus" and "srcState" that Correspond to "srcType"**

<i>srcType</i>	<i>srcStatus</i>	<i>srcState</i>
"PCI pwr" = 1	"Unk" = 0 (unknown) "On" = 1 "Off" = 2	"" = 0
"PCI aux" = 2	"Unk" = 0 (unknown) "On" = 1 "Off" = 2	"" = 0
"Battery" = 3	"<value> %charged" (value=0 if unknown)	"Unknown" = 0 "Charging" = 1 "Discharging" = 2 "Charged" = 3 "Not-charging" = 4 "Disconnected" = 5
"Ext pwr" = 4	"Unk" = 0 (unknown) "On" = 1 "Off" = 2	"Unknown" = 0 "Connected" = 1 "Disconnected" = 2

**Examples:**

- racadm getsensors**  

```

1 Temperature RAC III TEMP          37 degrees-C High-critical
Temperature
2 Voltage RAC III BATT VOLT   c6 Voltage   High-warning Battery
100% Charged
3 Voltage RAC III EXTPWR VOLT eb Voltage   Normal       Ext pwr
ON Connected
4 Voltage RAC III +12PCI VOLT bf Voltage   Disabled      PCI pwr
Unk
5 Voltage RAC III -12PCI VOLT ae Voltage   Disabled      PCI pwr
Unk
6 Voltage RAC III +5PCI VOLT c2 Voltage   Disabled      PCI pwr
Unk
7 Voltage RAC III AUXPCI VOLT 00 Voltage   Disabled      PCI pwr
Unk

```

- `racadm getsensors -s 2`  
2 Voltage RAC III BATT VOLT c6 Voltage High-warning Battery  
100% Charged
- `racadm getsensors -A -s 5`  
5 2 "RAC -12PCI VOLT" ae 4 6 1 00 0
- `racadm getsensors -A -E -h`  
1 1 "RAC TEMP " 37 1 5 0 00 0 19 c9 7d 00 14 31 37 3ff 000  
0 0 3 3  
2 2 "RAC BATT VOLT " c6 4 3 3 64 3 a0 69 b1 5a 7e c6 db 313 000  
0 4 5 5  
3 2 "RAC EXTPWR VOLT" eb 4 1 4 01 1 e6 c0 ff 98 df ee ff 2fc 000  
0 4 4 4  
4 2 "RAC +12PCI VOLT" bf 4 6 1 00 0 c0 b8 c8 b3 b8 c8 cc 18f 000  
0 4 3 3  
5 2 "RAC -12PCI VOLT" ae 4 6 1 00 0 ae aa b2 a8 aa b2 b4 3bb 000  
d 4 2 2  
6 2 "RAC +5PCI VOLT " c2 4 6 1 00 0 c0 b6 c9 ad b6 c9 d3 2fc 000  
0 4 4 4  
7 2 "RAC AUXPCI VOLT" 00 4 6 1 00 0 bf ae d1 a2 ae d1 dc 354 000  
0 4 6 6

## getssninfo

Table 1-21 describes the `getssninfo` subcommand.

**Table 1-21. getssninfo Subcommand**

Subcommand	Definition
<code>getssninfo</code>	Retrieves session information for one or more currently active or pending sessions from the Session Manager's session table.

## Synopsis

```
racadm getssninfo [-A] [-u <username> | *]
```

## Description

The `getssninfo` command returns a list of currently active or pending users and optionally includes summary session table information. The summary information provides the total number of sessions in each of the defined Session Manager states:

- Free
- Preliminary
- Unvalidated
- Valid
- Invalid

## Input

Table 1-22 describes the `getssninfo` subcommand options.

**Table 1-22. getssninfo Subcommand Options**

Option	Description
-A	The -A option eliminates the printing of data headers.
-u	The -u <i>&lt;username&gt;</i> user name option limits the printed output to only the detail session records for the given user name. If an "*" symbol is given as the user name, all users are listed. Summary information is not printed when this option is specified.

## Output

None

## Examples

- `racadm getssninfo`  
Session table summary status:  
1 VALID  
15 AVAILABLE

Table 1-23 provides an example of output from using the `racadm getssninfo` command.

**Table 1-23. getssninfo Subcommand Output Example**

Type	User	IP Address	Login Date/Time	Consoles
Web	RAC	root 143.166.174.19	Thu, 06 Mar 2003 10:32:39 GMT-06:00	NONE

- `racadm getssninfo -A`

```
1 15
```

```
"Web" "RAC\root" 143.166.174.19 "Thu, 06 Mar 2003 10:32:39 GMT-06:00" "NONE"
```

- `racadm getsninfo -A -u *`

```
"Web" "RAC\root" 143.166.174.19 "Thu, 06 Mar 2003 10:32:39 GMT-06:00" "NONE"
```

## getsysinfo

Table 1-24 describes the `getsysinfo` subcommand.

**Table 1-24. getsysinfo**

Command	Definition
<code>getsysinfo</code>	Displays RAC information, system information, and watchdog status information.

### Synopsis

```
racadm getsysinfo [-d] [-s] [-w] [-A]
```

### Description

The `getsysinfo` command returns information about watchdog status, depending on the command options.

### Input

Table 1-25 describes the `getsysinfo` subcommand options.

**Table 1-25. getsysinfo Subcommand Options**

Option	Description
<code>-w</code>	Displays watchdog information.
<code>-A</code>	Eliminates the printing of headers/labels.

If the `-w` option is not specified, then these three options are used as defaults.

### Output

The following data element is output as a string:

```
Watchdog information/recovery action
```

Enumeration values or bitmaps are defined for these elements. When the `-A` (API) option is included on the command, the enumeration/bit value of the element is listed in the output. Otherwise, the enumeration or bit value is mapped to a string before being output.

The bulleted items listed in Table 1-26 define the mapping of values to strings.

**Table 1-26. Watchdog Recovery Action Number Descriptions**

Value	Description
Watchdog recovery action	An enumeration defines the meaning of this value: <ul style="list-style-type: none"> <li>• 0 = no-action</li> <li>• 1 = hard-reset</li> <li>• 2 = power-down</li> <li>• 3 = power-cycle</li> </ul>

### Examples

- `racadm getsysinfo -A -w -s`  

```
"123456" "PowerEdge 1650" "A08" "EF23VQ-0023" "" 0x100
"Server1"

"Microsoft Windows 2000 version 5.0, Build Number 2195, Service
Pack 2" "1.60"

"Watchdog Info:" 2 39 40
```
- `racadm getsysinfo -w -s`  

```
System Information:
System ID           = 123456
System Model        = PowerEdge 1650
BIOS Version        = A08
Asset Tag           = EF23VQ-0023
Service Tag         = 2MP9Z01
Hostname            = Server1
OS name             = Microsoft Windows 2000 version 5.0,
                    Build 2195 Service Pack 2
ESM Version         = 1.60

Watchdog Information:
Recovery Action      = Power Down
Present countdown value = 36
```

Initial countdown value = 40

## setrtime/getrtime

Table 1-27 describes the **setrtime** and **getrtime** subcommands.

**Table 1-27. setrtime/getrtime**

Subcommand	Definition
setrtime	Sets the time on the controller.
getrtime	Displays the time from the controller.



**NOTICE:** You must reconfigure the controller time when you perform a firmware update or a RAC reset, or when you lose AC power to the chassis.

### Synopsis

```
racadm setrtime -s
racadm setrtime -u <utctime> [-o <offset>]
racadm setrtime -d yyyyymmddhhmmss.mmmmmms<offset>
racadm getrtime [-u] [-d]
```

### Description

The **setrtime** subcommand sets the time on the controller. The time can be specified in three different ways and the default is **-s**.

**-s** – Available only if executed from the remote **racadm** utility (not serial). The time is taken from the system time that **racadm** is executing.

The **racadm setrtime** subcommand calls the **librac** function "drsGetOsTime(UINT32 \*utcTime, UINT32 \*utcOffset) to get the time and offset from the operating system and then executes the command:

```
setgmt -u <utctime> -o <offset>
```

**-u** – The time is specified by the user in UTC-seconds since 1/1/1970.

**-o** – The offset used with **-u** is the seconds offset from GMT (a signed value).

**-d** – The local time is specified as a string: *yyyyymmddhhmmss.mmmmmms<offset>*, where *yyyy* is a 4-digit year.

- *mm* is the month.
- *dd* is the day.
- *hh* is the hour (24-hour clock).
- *mm* is the minute.

- *ss* is the second.
- *mmmmmm* is the number of microseconds.
- *s* is a "+" or "-", indicating the sign of the offset value.
- *<offset>* equals the GMT – local time (in minutes).

For example, Monday, May 25, 2003, at 1:30:15 PM EST would be represented as:  
20030525133015.0000000-300.

### Description for getractive

The **getractive** subcommand displays the time in one of the two formats:

- *u* – The UTC hexadecimal value followed by the offset in signed decimal (default).
- *d* – The *yyyymmddhhmmss.mmmmmmsoff* string with no option is displayed the same as the UNIX<sup>®</sup> date command.

### Output

The **setractive** subcommand returns without output if successful and the **getractive** subcommand displays the output on one line.

### Restrictions

The *-s* option for **setractive** is only available as a **racadm** subcommand.

### setniccfg/getniccfg

Table 1-28 describes the **setniccfg** and **getniccfg** subcommands.

**Table 1-28. setniccfg/getniccfg**

Subcommand	Definition
setniccfg	Sets the IP configuration for the controller.
getniccfg	Displays the current IP configuration for the controller.



**NOTE:** The terms network adapter and Ethernet management port may be used interchangeably.

### Synopsis

```
racadm setniccfg -d
racadm setniccfg -s [<ipAddress> <netmask> <gateway>]
racadm setniccfg -o
racadm getniccfg
```

**Description**

The `setniccfg` subcommand sets the controller IP address.

- The `-d` option enables DHCP for the Ethernet management port (default is DHCP enabled).
- The `-s` option enables static IP settings. The IP address, netmask, and gateway can be specified. Otherwise, the existing static settings are used.
- The `-o` option disables the Ethernet management port completely.

`<ipAddress>`, `<netmask>`, and `<gateway>` are entered as dot-separated strings.

**Description**

The `getniccfg` subcommand displays the current Ethernet management port settings

**Output**

The `setniccfg` subcommand returns without output if successful. The `getniccfg` subcommand output displays the following information:

```
Network adapter = Enabled/Disabled
DHCP = Enabled/Disabled
Static IP Settings: <ipAddress> <netmask> <gateway>
Current IP Settings: <ipAddress> <netmask> <gateway>
```

**getsvctag**

Table 1-29 describes the `getsvctag` subcommand.

**Table 1-29. getsvctag**

Subcommand	Definition
<code>getsvctag</code>	Displays a service tag.

**Synopsis**

```
racadm getsvctag [-m <module>]
```

**Description**

The `getsvctag` subcommand is used to display the Service Tag for a module or all modules with unique Service Tags. The default is all Service Tag modules.

**Example:**

Type `getsvctag` at the command prompt. The output is displayed as follows:

```
chassis Y76TP0G
server-1 Y76TP0H
server-n TR3H33P
```

Both commands return 0 on success and nonzero on errors.

## setassettag/getassettag

Table 1-30 describes the `setassettag` and `getassettag` subcommands.

**Table 1-30. setassettag/getassettag**

Subcommand	Definition
<code>setassettag</code>	Sets the specified asset tag.
<code>getassettag</code>	Displays asset tags.

### Synopsis

```
racadm setassettag -m <module> <assetTag>
```

```
racadm getassettag [-m <module>]
```

### Description for setassettag

The `setassettag` subcommand is used to set the asset tag for the specified module. Only the chassis asset tag can be set or displayed.

`<module>` has the following legal value: chassis.

`<assetTag>` is the *N*-byte ASCII asset tag where *N* must be less or equal to 10 bytes.

### Description for getassettag

The `getassettag` subcommand is used to display the asset tag for a module or all modules with unique asset tags. The default is all modules (that have asset tags).

### Example:

Type `getassettag` at the command prompt. The output is displayed as follows:

```
chassis 7837339-33.
```

Both subcommands return 0 on success and nonzero on errors.

## localauthenable

Table 1-31 describes the `localauthenable` subcommand.

**Table 1-31. localauthenable**

Subcommand	Definition
<code>localauthenable</code>	Enables or disables the local operating system authentication.

**Synopsis**

```
racadm localauthenable [<TRUE|FALSE>]
```

**Description**

The **localauthenable** subcommand determines if the RAC out-of-band GUI allows users to authenticate a user session using the local operating system user or the user credentials stored in the RAC database. If the setting is modified, the user must close and reopen the browser to see the change.

**Input**

TRUE — Enables operating system authentication.

FALSE — Disables operating system authentication.

**Output**

If you do not enter an argument, the **localauthenable** current state is returned.

**racdump**

Table 1-32 describes the **racdump** subcommand.

**Table 1-32. racdump**

Subcommand	Definition
racdump	Displays status and general RAC information.

**Synopsis**

```
racadm racdump
```

**Description**

The **racdump** subcommand provides a single command to get a dump, status, and general RAC board information with a single command.

The following commands are executed as a result of the single **racdump** subcommand:

- getsysinfo
- coredump
- memmap
- netstat
- getssninfo
- getsensors

## Input

None

## Output

The output of the individual commands are displayed.

## racreset

Table 1-33 describes the **racreset** subcommand.

**Table 1-33. racreset**

Subcommand	Definition
racreset	Resets the RAC.



**NOTICE:** You must wait until the RAC reset is completed before issuing another command. If the RAC reset is not completed, you may receive the following error: `racadm <command name>`  
Transport: ERROR: (RC=-1)

## Synopsis

```
racadm racreset [hard | soft | graceful] [delay]
```

## Description

The **racreset** subcommand issues a reset to the RAC. The user is allowed to select how many seconds of delay occur before the reset sequence is started. The reset event is written into the RAC log.

The default option is **soft**. If you do not enter an option, `racadm` waits three seconds and then runs the **soft** option with the **racreset** subcommand.

## Input



**NOTICE:** You must reboot your system after performing a hard reset of the RAC as described in the Table 1-34.

Table 1-34 describes the **racreset** subcommand options.

**Table 1-34. racreset Subcommand Options**

Option	Description
hard	A <i>hard</i> reset resets the entire RAC and is as close to a power-on reset as can be achieved using software. The RAC log, database, and selected daemons are shutdown gracefully prior to the reset. A hard reset should be considered as a final effort. PCI configuration is lost.

**Table 1-34. racreset Subcommand Options**

Option	Description
soft	A <i>soft</i> reset is a processor and processor subsystem reset that resets the processor core to restart the software. PCI configurations are preserved. The RAC log, database, and selected daemons are shutdown gracefully prior to the reset.
graceful	A <i>graceful</i> reset is the same as a soft reset.
<delay>	The user is allowed to select how many seconds of delay occur before the reset sequence is started. A valid delay entry is between 1-60 seconds. The default is 3 seconds.

**Output**

None

**Examples**

- `racadm racreset soft 1`  
Start the RAC soft reset sequence in 1 second.
- `racadm racreset soft 20`  
Start the RAC soft reset sequence after 20 seconds.

**racresetcfg**Table 1-35 describes the `racresetcfg` subcommand.**Table 1-35. racresetcfg**

Subcommand	Definition
<code>racresetcfg</code>	Resets all database configuration parameters to default values; equivalent to a soft reset.

**Synopsis**`racadm racresetcfg`**Description**

The `racresetcfg` command removes all database property entries that have been configured by the user. The database has default properties for all entries that are used to restore the card back to its original default settings. After resetting the database properties, the RAC resets automatically.



**NOTICE:** Before using this command, ensure that you want to restore your database to its original default state with default user `root` and default password `calvin`.



**NOTE:** After issuing a `racresetcfg` subcommand, stop and then restart the following services: Server Agent, Server Agent Event Monitor, and SNMP. See the *Dell OpenManage Server Administrator's User's Guide* for information on stopping and restarting the services in your operating system.

### Input

None

### Output

None

## serveraction

Table 1-36 describes the `serveraction` subcommand.

**Table 1-36. serveraction**

Subcommand	Definition
serveraction	Executes a managed system reset or power-on/off/cycle.

### Synopsis

```
serveraction [-d <delay>][-w <cycleWait>] <action>
```

### Description

The `serveraction` command provides an interface to control system reset and power control. Table 1-37 describes the `serveraction` subcommand option values.

**Table 1-37. serveraction Subcommand Options**

String	Definition
<action>	Specifies the action. The options for the <action> string are: <ul style="list-style-type: none"> <li>• <code>powerdown</code> – Powers down the server module.</li> <li>• <code>powerup</code> – Powers up the server module.</li> <li>• <code>powercycle</code> – Issues a power-cycle to the server module.</li> <li>• <code>hardreset</code> – Issues a hard reset to the server module.</li> <li>• <code>graceshutdown</code> – Powers down the server module gracefully.</li> <li>• <code>gracepowercycle</code> – Powers down the server module gracefully (same as the <code>graceshutdown</code> option)</li> </ul>
<delay>	Specifies the time in seconds after the command is received before the action is executed. The default is 1 second.

### Output

The `serveraction` command returns without output if successful.

## getmodinfo

Table 1-38 describes the `getmodinfo` command.

**Table 1-38. getmodinfo**

Command	Definition
<code>getmodinfo</code>	Displays module configuration and status information. See Table 1-39.

### Synopsis

```
getmodinfo [-m <module>] [-A] [-G <generation>]
```

### Description

The `getmodinfo` command displays the following information about the major modules in a chassis:

- Module name
- Presence status
- Power status
- Health
- Module service tag

`<module>` has the following values:

- chassis
- server-`<n>` (n=1 to 6) // ie server-1
- switch-`<n>` (n =1,2)// ie switch-1
- smcard
- fan-`<n>` (n=1 to 4)//ie fan-1
- ps-`<n>` (n=1,2)//ie ps-2

The default is to display information about all of the major modules in the chassis. The `-A` option suppresses the header and outputs `<presence>`, `<pwrState>`, and `<health>` as a numeric character.

The `-G <generation>` option returns 1 if `<generation>` matches the current generation of the module resource table (indicating any changes since the last call). If the generation is new and the `-G` option was used, the following line is printed before the rest of the module information:

```
"Generation: <n> " where <n> is the generation number.
```

A *<generation>* value of 0 always causes a mismatch and thus new output.

Table 1-39 shows an example of the `getmodinfo` command output.

**Table 1-39. getmodinfo Output Example**

Generation:					
34323					
#<group>	<presence>	#<module>	<pwrState>	<health>	<svcTag>
1	Chassis	Present	ON	Warning	4TY9W29
	Fan-1	Present	ON	Failed	4TY9W29
	Smcard	Present	ON	OK	4TY9W29
2	Server-1	Present	OFF	na	88P9A46

## Output

The `getmodinfo` command returns without output if successful, and prints a line of output for each module specified.

## getraclog

Table 1-40 describes the `getraclog` command.

**Table 1-40. getraclog**

Command	Definition
<code>getraclog -i</code>	Displays the number of entries in the RAC log.
<code>getraclog</code>	Displays the RAC log entries.

## Synopsis

```
getraclog -i
```

```
getraclog [-A] [-c count] [-d delay-seconds] \[-s start-record] [-v] [-V] [-m]
```

## Description



**NOTE:** The command name and the `racadm` subcommand names may be different. This is normal.

The `getraclog -i` command displays the number of entries in the RAC log.

The following are options for the `getraclog` command to read entries:

- `-A` – Provides API-formatted output (no header).

- **-c** – Provides the maximum count of entries to be returned.
- **<blank>** – Displays the entire log; racadm and serial only (default).
- **-d** – Provides the number of seconds to delay the recording of any new log entries.
- **-s** – Provides the associated number of the first displayed entry (default = 0 [list begins with the first RAC log entry]).
- **-v** – Provides "verbose" output.
- **-V** – Provides "Very verbose" output.
- **-m** – Displays 24 rows at a time, and queries for more (such as the UNIX<sup>®</sup> **more** command).

## Output

One line of output is displayed for each RAC log entry.

## Restrictions

The output buffer size is too big for execution across IPMI transport.

## clrraclog

### Synopsis

```
clrraclog
```

### Description



**NOTE:** The command name and the racadm subcommand names may be different. This is normal.

The **clrraclog** command completely clears the RAC log. A single entry is made to indicate the user and time that the log was cleared.

## getsel

Table 1-41 describes the **getsel** command.

**Table 1-41. getsel**

Command	Definition
getsel -i	Displays number of entries in the System Event Log.
getsel	Displays SEL entries.

## Synopsis

```
getsel -i
```

```
getsel [-E] [-R] [-A] [-c count] [-d delay-seconds]\[-s count] [-v] [-V] [-m]
```

## Description

The **getsel -i** command displays the number of entries in the SEL.

The **clrsel** command completely clears the SEL.

The following **getsel** options (without the **-i** option) are used to read entries.

**-A** – Provides API-formatted output (no header).

**-c** – Provides the maximum count of entries to be returned.

**<blank>** – Default is to display the entire log) racadm and serial only.

**-d** – Provides the number of seconds to delay the recording of any new log entries.

**-s** – Provides the number of records to skip before returning entries (default=0).

**-v** – Provides "verbose" output.

**-V** – Provides "Very verbose" output.

**-E** – Places the 16 bytes of raw SEL at the end of each line of output as a sequence of hex values.

**-R** – Only the raw data is printed.

**-m** – Displays 24 rows at a time, and queries for more (such as the UNIX **more** command).

## Output

One line of output is displayed for each SEL entry.

## clrsel

### Synopsis

```
clrsel
```

### Description

The **clrsel** command completely clears the System Event Log entries.

### connect

Table 1-42 describes the **connect** command.

**Table 1-42. connect**

Command	Definition
connect	Connects to a server module or switch serial port.

**Synopsis**

```
connect [-b] [-u] <module>
```

**Description**

The **connect** command causes the management serial port to connect to the serial port associated with the module specified by *<module>*.

This command is used to connect the system management serial port to a server module serial port, or a switch serial port.

The *<module>* parameter has the following values for the connect command:

- server-*<n>* – (n=1 to 6) For example, `server-1`.
- switch-*<n>* – (n =1, 2) For example: `switch-1`.

The following options are available for the **connect** command:

**-b** – Connects in binary mode (normally used for downloading or uploading binary files).

**-u** – Connects in unsecure mode. Use unsecure mode if you need to disconnect your serial cable from the management serial port without causing the controller to log you off of the session.

If options are not specified, the default connection is in text and secure mode.



**NOTE:** Because binary and unsecure mode is not supported, the **-b** and **-u** options cannot be used together.

**Guidelines for Terminating a Connection**

- When the connection is in text mode, press `<Enter><~><.>` to terminate the connection.
- When the connection is in binary mode, the break sequence will no longer function. In this case, the connection can only be terminated by host terminal disconnection.
- When the connection is in secure mode, any host-terminal-to-management-serial-port disconnection causes the connection to terminate and causes the user to log out from the serial command shell.
- Anytime a "~" character is seen following a `<CR>` (carriage return), the "~" is not forwarded to the port until the next character after the "~" is read. If the next character is a ".", then the connection is broken (and the "~." is never transmitted to the port).

- If the next character (after the "~") is not a ".", then both the "~" and the next character are transmitted to the port. Therefore, any communication to the server modules, switches, or KVM must not send the character sequence: <CR><~><. > or else the connection will be terminated.

## Output

The connect command output prints the following single output line and then connects to the port:

```
Connected to <module> ..."
```

When the connection is terminated (with the "<CR>~." sequence in text mode), the following output is printed:

```
Connection to <module> terminated."
```

```
<CMD_PROMPT>
```

and a command prompt is printed.

## gettracelog

Table 1-43 describes the `gettracelog` subcommand.

**Table 1-43. gettracelog**

Command	Definition
<code>gettracelog -i</code>	Displays the number of entries in the RAC trace log.
<code>gettracelog</code>	Displays the RAC trace log.

## Synopsis

```
gettracelog -i
```

```
gettracelog [-m]
```

## Description

The `gettracelog -i` command displays the number of entries in the RAC trace log. The `gettracelog` (without the `-i` option) command is used to read entries.

The `-m` option displays 24 rows at a time, and queries for more (such as the UNIX `more` command).

## Output

One line of output is displayed for each trace log entry.

## setrac

Table 1-44 describes the **setrac** subcommand.


**Table 1-44. setrac**

Subcommand	Definition
setrac	Sets RAC parameters from the managed system. When used remotely, sets RAC parameters from the management station.

## Synopsis

```
racadm setrac [-t -h -o -T -d]
```

## Description

 **NOTE:** The **racadm setrac** command cannot be used remotely.

The **setrac** command sets time, managed system name, operating system name, or operating system type from the managed system to the RAC. If options are not entered, all four parameters are set. The **-d** option allows the user to display the parameters only without actually writing them to the RAC firmware. Any combination of options, or no options, may be entered.

## Input

Table 1-45 describes the **setrac** subcommand options.

**Table 1-45. setrac Subcommand Options**

Option	Description
-t	Gets the "UTC time" from the managed system and sets the RAC with this time.
-h	Gets the "Hostname" from the managed system and writes it to the RAC. This parameter is available for viewing using the <b>getsysinfo</b> command, and under the object ID interface using <b>config/getconfig</b> as <b>[ifcRacManagedNodeOs] ifcRacMnOsHostname</b> .
-o	Gets the "OS Name" from the managed system and writes it to the RAC. This parameter is available for viewing under the object ID interface using <b>config/getconfig</b> as <b>[ifcRacManagedNodeOs] ifcRacMnOsName</b> .
-T	Gets the "OS Type" from the managed system and writes it to the RAC. This parameter is available for viewing using the <b>getsysinfo</b> command and under the object ID interface using <b>config/getconfig</b> as <b>[ifcRacManagedNodeOs] ifcRacMnOsOsType</b> .

**Table 1-45. setrac Subcommand Options**

<b>-d</b>	The <b>-d</b> , or <b>display</b> option, allows the user to get the "UTC time," "Hostname," "OS Name," and "OS Type" from the managed system and display these items only. Parameters are not written to the RAC. If the <b>-d</b> option is entered along with other options, then those parameters are displayed only.
-----------	---

### Output

A message is displayed indicating the values obtained from the managed system, and if they are being written to the RAC or only displayed.

### Examples

- `racadm setrac -d` – Only the parameter values are displayed.
- `racadm setrac -t -h` – The UTC time and managed system name are taken from the managed system and written to the RAC.

## sslsrsgen

Table 1-46 describes the `sslsrsgen` subcommand.

**Table 1-46. sslsrsgen**

Subcommand	Description
<code>sslsrsgen</code>	Generates and downloads the SSL CSR.

### Synopsis

```
racadm sslsrsgen [-g] [-u] [-f <filename>]
```

```
racadm sslsrsgen -s
```

### Description

The `sslsrsgen` subcommand is used to generate the CSR and download it to a file, *<filename>*.

### Options



**NOTE:** The `-u` and `-f` options are not supported for the serial/telnet console.

Table 1-47 describes the `sslsrsgen` subcommand options.

**Table 1-47. sslsrsgen Subcommand Options**

Option	Description
<code>-g</code>	Generates a new CSR.
<code>-s</code>	Returns the status of a CSR generation process (generation in progress, active, or none).

**Table 1-47. sslcsrgen Subcommand Options**

<b>-u</b>	Uploads the CSR to the filename specified by the <b>-f</b> option.
<b>-f</b>	Specifies the filename of the location, <i>&lt;filename&gt;</i> , where the CSR will be downloaded.

 **NOTE:** If the **-f** option is not specified, the filename defaults to **sslcsr** in your current directory.

If options are not specified, the default is **-g -u**. The **-g -u** options (together) cannot be used with the **-s** option. The **-f** option must be used with the **-u** option.

The **sslcsrgen -s** subcommand returns one of the following status codes:

0x00000000 – CSR was generated successfully.

0x40040014 – CSR does not exist.

0x40040006 – CSR generation in progress.

0x40040009 – Key size is not supported.

The **sslcsrgen -u** subcommand downloads the CSR from the RAC by FTP. This command returns 0 when successful and returns a nonzero number when unsuccessful.

### Restrictions

The **sslcsrgen** command can only run on a system that has the managed system software installed.

### Example

```
racadm sslcsrgen -s
```

or

```
racadm sslcsrgen -g -u -f c:\csr\csrtest.txt
```

### sslcertupload

Table 1-48 describes the **sslcertupload** subcommand.

**Table 1-48. sslcertupload**

Subcommand	Description
sslcertupload	Downloads a CA certificate to the RAC.

### Synopsis

```
racadm sslcertupload -t <type> [-f <filename>]
```

### Options

Table 1-49 describes the **sslcertupload** subcommand options.

**Table 1-49. sslcertupload Subcommand Options**

Option	Description
-t	Specifies the type of certificate to upload, either the CA certificate or server certificate. 0x1 = server certificate 0x2 = CA certificate
-f	Specifies the filename, <filename>, to be uploaded.



**NOTE:** If the -f option is not specified, the filename defaults to **sslcert** in your current directory.

The **sslcertupload** command returns 0 when successful and returns a nonzero number when unsuccessful.

### Restrictions

The **sslcertupload** command can only run on a system that has the managed system software installed.

### Example

```
racadm sslcertupload -t 0x1 -f c:\cert\cert.txt
```

### sslcertview

Table 1-50 describes the **sslcertview** subcommand.

**Table 1-50. sslcertview**

Subcommand	Description
sslcertview	Displays a CA certificate or server certificate that is located in the RAC.

### Synopsis

```
racadm sslcertview -t <type> [-A]
```

### Options

Table 1-51 describes the **sslcertview** subcommand options.

**Table 1-51. sslcertview Subcommand Options**

Option	Description
--------	-------------

**Table 1-51. sslcertview Subcommand Options**

<b>-t</b>	Specifies the type of certificate to upload, either the CA certificate or server certificate.  0x1 = server certificate 0x2 = CA certificate
<b>-A</b>	Prevents printing headers/labels.

**Output Example 1**

For the `racadm sslcertview -t 1` subcommand, you receive output similar to the following example, where **C** is the country, **CN** is the common name, **O** is the organization, **OU** is the organizational unit, **L** is the locality, **S** is the state, and **E** is the e-mail address:

```
certificate type=1
serial number=00
key size=1024
valid from=DSU+12:34:31
valid to=DSU+15:34:31
subject.C=US
subject.CN=RMC Default Certificate
subject.O=Dell Inc.
subject.OU=BVS
subject.L=Round Rock
subject.S=Texas
subject.E=john@dell.com
issuer.C=US
issuer.CN=RMC Default Certificate
issuer.O=Dell Inc.
issuer.OU=BVS
issuer.L=Round Rock
issuer.S=Texas
issuer.E=john@dell.com
```

**Output Example 2**

For the `racadm sslcertview -t 1 -A` subcommand, you receive output similar to the following example:

```
1
00
1024
DSU+12:34:31
DSU+15:34:31
US
RMC Default Certificate
```

Dell Inc.  
BVS  
Round Rock  
Texas  
john@dell.com  
US  
RMC Default Certificate  
Dell Inc.  
BVS  
Round Rock  
Texas  
john@dell.com

## **syncvncpassword**

Table 1-52 describes the `syncvncpassword` subcommand.

**Table 1-52. syncvncpassword**

<b>Subcommand</b>	<b>Definition</b>
<code>syncvncpassword</code>	Sets the managed system VNC server password from the RAC.

### **Synopsis**

```
racadm syncvncpassword
```

### **Description**

The `syncvncpassword` subcommand reads the VNC password from the RAC firmware and sets the managed system's RAC VNC server to the same value.

### **Input**

None

### **Output**

None

## **racadm Test Subcommands**

This section provides subcommands that are used to test the RAC alert functions.

### **testalphapage**

Table 1-53 describes the `testalphapage` subcommand.

**Table 1-53. testalphapage**

Subcommand	Description
testalphapage	Sends an alphanumeric test page to a user.

**Synopsis**

```
racadm testalphapage <-i <index> | -u <username>>
```

 **NOTE:** The DRAC III is the only RAC that supports modem capabilities.

**Description**

The `testalphapage` subcommand forces the RAC to send an alphanumeric page through the RAC modem to a user.

**Input**

Table 1-54 describes the `testalphapage` subcommand options.

**Table 1-54. testalphapage Subcommand Options**

Option	Description
-u	Specifies the user who receives the page. The necessary properties must be set up to correctly page out.
-i	Specifies the index of the user.

**Output**

None

**testemail**

Table 1-55 describes the `testemail` subcommand options.

**Table 1-55. testemail**

Subcommand	Description
testemail	Tests an e-mail alert.

**Synopsis**

```
racadm testemail <-i <index> | -u <username>>
```

**Description**

The `testemail` subcommand forces the RAC to send an e-mail over the RAC network adapter.

## Input

Table 1-56 describes the **testemail** subcommand options.

**Table 1-56. testemail Subcommand Options**

Option	Description
-u	Specifies the user who receives the e-mail. The necessary properties must be setup to correctly page out.
-i	Specifies the index of the user.

## Output

None

## testnumpage

Table 1-57 describes the **testnumpage** subcommand.

**Table 1-57. testnumpage**

Subcommand	Description
testnumpage	Sends a numeric test page to a user.

## Synopsis

```
racadm testnumpage <-i <index> | -u <username>>
```

## Description

The **testnumpage** subcommand forces the RAC to send a numeric page through the RAC modem to a user.



**NOTE:** The DRAC III is the only RAC that supports modem capabilities.

## Input

Table 1-58 describes the **testnumpage** subcommand options.

**Table 1-58. testnumpage Subcommand Options**

Option	Description
-u	Specifies the user who receives the page. The necessary properties must be setup to correctly page out.
-i	Specifies the index of the user.

**Output**

None

**testtrap**

Table 1-59 describes the **testtrap** subcommand.

**Table 1-59. testtrap**

Subcommand	Description
testtrap	Tests an SNMP trap.

**Synopsis**

```
racadm testtrap -i <index>
```

**Description**

The **testtrap** subcommand forces the RAC to send an SNMP trap over the RAC network adapter.

**Input**

Table 1-60 describes the **testtrap** subcommand options.

**Table 1-60. testtrap Subcommand Options**

Option	Description
-i	Specifies the index of the trap.

**Output**

None

## Using the racadm Utility to Configure the RAC

The fastest method to configure the RAC is to use Server Administrator. If Server Administrator is unavailable, or if you prefer command-line or script configuration, you can also configure the RAC by using the racadm utility. The racadm utility is installed along with the RAC agents on the managed system.

To configure multiple RACs to contain the same user configuration settings, you can do either of the following:

- Using the racadm utility examples in this section as a guide, create a batch file of **racadm** commands, and then execute the batch file on each managed system.

- Create the RAC configuration file as described in "racadm Subcommand Man Pages;" then execute the **racadm config** subcommand on each managed system using that same configuration file.

## Locating the Proper User Index

The RAC allows up to 16 users to be configured into the RAC property database. Before manually adding the RAC user, you need to know which, if any, users exist. If the RAC is new, or the **racadm racresetcfg** command has been run, then the only user is **root** with the password **calvin**. The **racresetcfg** subcommand resets the RAC back to the original defaults.



**NOTICE:** Use caution when using this command because *all* configuration parameters are reset to the original defaults; any previous changes are lost.



**NOTE:** Users can be added and deleted over time, so it is very possible that users on the RAC do not have the same index number as the same user on a different RAC.

To find out if a user exists, enter the following command at the command prompt, once for each index of 1–16.

```
racadm getconfig -g cfgUserAdmin -i <index>
```



**NOTE:** An alternate method to obtain this information would be to enter `racadm getconfig -f <myfile.cfg>`, then view or edit the **myfile.cfg** file, which includes all RAC configuration parameters.

Over 20 parameters and object IDs are displayed along with their current values. The two objects of interest are:

```
# cfgUserAdminIndex=XX
cfgUserAdminUserName=
```

If the **cfgUserAdminUserName** object has no value, that index number, which is indicated by the **cfgUserAdminIndex** object, is available for use. If a name appears after the "=", that index is taken by that user name.



**NOTE:** When you manually add or remove a user with the **racadm config** subcommand, you *must* specify the index with the **-i** option. Observe that the **cfgUserAdminIndex** object displayed in the previous example contains a '#' character. Also, if you use the **racadm config -f racadm.cfg** command to specify any number of groups/objects to write, the index cannot be specified. A new user is added to the first available index. This behavior allows more flexibility in configuring multiple RACs with the same settings.

## Adding the RAC User Without Alert Capabilities

To add a simple user without any alert information, first locate an available user index by performing the steps in "Locating the Proper User Index." Next, enter the following two command lines with the new user name and password:

```
racadm config -g cfgUserAdmin -o cfgUserAdminUserName -i <index>
<username>
```

```
racadm config -g cfgUserAdmin -o cfgUserAdminPassword -i <index>
<password>
```

#### Example:

```
racadm config -g cfgUserAdmin -o cfgUserAdminUserName -i 2 john
racadm config -g cfgUserAdmin -o cfgUserAdminPassword -i 2 123456
```

A user name "john" with the password of "123456" is created. This user name and password can now be used to log into the Web-based remote access interface. You can verify this using either of the following two commands:

```
racadm getconfig -u john
racadm getconfig -g cfgUserAdmin -i 2
```

### Deleting a RAC User

All users must be deleted manually. You cannot delete users by specifying them in a `racadm.cfg` file.


To delete the user "john" created in the previous example, enter the following command line:

```
racadm config -g cfgUserAdmin -o cfgUserAdminUserName -i <index>
""
```

A NULL string (two " characters[""]) indicates to the RAC to delete the index for the specified group.

### Adding a RAC User With Alerting Capabilities

To add a RAC user that is able to receive e-mail and SNMP traps, first locate an available RAC user index by performing the steps in "Locating the Proper User Index." The following example has an available user index at index 2.

 **NOTE:** See "RAC Property Database Group and Object Definitions" for details about each specific object.

```
racadm config -g cfgUserAdmin -o cfgUserAdminUserName -i 2 john
racadm config -g cfgUserAdmin -o cfgUserAdminPassword -i 2 123456
racadm config -g cfgUserAdmin -o cfgUserAdminPageAlphaNumber -i 2
9,18002506325
racadm config -g cfgUserAdmin -o cfgUserAdminPageAlphaPagerId -i 2
415-0722
racadm config -g cfgUserAdmin -o cfgUserAdminPageAlphaBaudRate -i
2 0x4B0
racadm config -g cfgUserAdmin -o cfgUserAdminPageAlphaProtocol -i
2 8N1
```

```

racadm config -g cfgUserAdmin -o cfgUserAdminPageAlphaCustomMsg -i
2 "RAC Alpha Page test"

racadm config -g cfgUserAdmin -o cfgUserAdminPageAlphaEnable -i 2
1

racadm config -g cfgUserAdmin -o cfgUserAdminPageNumericNumber -i
2 9,18774150722

racadm config -g cfgUserAdmin -o cfgUserAdminPageNumericMessage -i
2 "111-222-3333"

racadm config -g cfgUserAdmin -o cfgUserAdminPageNumericEnable -i
2 1

racadm config -g cfgUserAdmin -o cfgUserAdminPageModemInitString -
i 2 AT+GCI=B5

racadm config -g cfgUserAdmin -o cfgUserAdminEmailAddress -i 2
"<email address>"

racadm config -g cfgUserAdmin -o cfgUserAdminEmailCustomMsg -i 2
"RAC Alert Email Test"

racadm config -g cfgUserAdmin -o cfgUserAdminEmailEnable -i 2 1

racadm config -g cfgUserAdmin -o
cfgUserAdminAlertFilterRacEventMask -i 2 0x0

racadm config -g cfgUserAdmin -o
cfgUserAdminAlertFilterSysEventMask -i 2 0x0

racadm config -g cfgTraps -o cfgTrapsSnmpCommunity -i 2 public

racadm config -g cfgTraps -o cfgTrapsEnable -i 2 1

racadm config -g cfgTraps -o cfgTrapsFilterRacEventMask -i 2 0x0

racadm config -g cfgTraps -o cfgTrapsFilterSysEventMask -i 2 0x0

racadm config -g cfgTraps -o cfgTrapsDestIpAddr -i 2 <SNMP trap
destination>

racadm config -g cfgDialupNetworking -o cfgDialupDialType 0x1

racadm config -g cfgOobSnmp -o cfgOobSnmpTrapsEnable 1

racadm config -g cfgRemoteHosts -o cfgRhostsSmtptServerIpAddr
143.166.224.254

racadm racreset

```

You can enter the commands manually, or run a batch file, or build a .cfg file using the command `racadm config -f racadm.cfg`. After doing so, you may want to test each of the alerts.

### Testing E-mail Alerting

E-mail alerting is enabled by the following command. A "0" disables this feature; a "1" enables it.

```
racadm config -g cfgUserAdmin -o cfgUserAdminEmailEnable -i 2 1
racadm testemail -i 2
```

### Testing SNMP Trap Alerting

SNMP traps are enabled by the following command. A "0" disables this feature; a "1" enables it.

```
racadm config -g cfgTraps -o cfgTrapsEnable -i 2 1
racadm testtrap -i 2
```

## Configuring RAC Network Properties

Enter the following command to get a list of the available network properties:

```
racadm getconfig -g cfgLanNetworking
```

If you want to use DHCP to obtain an IP address, you can use the command to write the object `cfgNicUseDhcp` to enable it. You may also enter a static IP address, netmask, and gateway.

The commands provide the same configuration functionality as the option ROM does at boot-up time when prompted to enter <Ctrl><d>. For more information about configuring network properties with the option ROM, see "Configuring RAC Network Properties."

The following is an example of the commands that may be entered to configure desired LAN network properties.

```
racadm config -g cfgLanNetworking -o cfgNicEnable 1
racadm config -g cfgLanNetworking -o cfgNicIpAddress 192.168.20.1
racadm config -g cfgLanNetworking -o cfgNicNetmask 255.255.255.0
racadm config -g cfgLanNetworking -o cfgNicGateway 192.168.20.1
racadm config -g cfgLanNetworking -o cfgNicUseDhcp 0
```

You must reset the RAC using the following command before the new configuration parameters will take effect:


```
racadm racreset
```




**NOTE:** If `cfgNicEnable` is set to 0, the RAC LAN is disabled even if DHCP is enabled.

## Locating the Proper Dial-In (PPP) User Index

The RAC allows up to 16 dial-in (PPP) users to be configured. Before manually adding a dial-in user, you need to know which, if any, dial-in users exist. If the RAC is new, or the `racadm racresetcfg` command has been run, dial-in users will not exist. Also, default dial-in users (the only default user and password are "root" and "calvin," which are RAC users) will not exist. The `racresetcfg` subcommand resets the RAC back to the original defaults.

 **NOTICE:** Use caution with this command because *all* configuration parameters are reset to original defaults; any previous changes are lost.

 **NOTE:** Dial-in users can be added and deleted over time, so it is possible that users on the RAC do not have the same index number as the same user on a different RAC.

To find out if a dial-in user exists, enter the following command, once for each index of 1 through 16.


```
racadm getconfig -p -g cfgDialinUsers -i <index>
```

 **NOTE:** An alternate method to obtain this information would be to enter `racadm getconfig -f <myfile.cfg>`, then simply view or edit `myfile.cfg`, which includes all RAC configuration parameters.

Five object IDs are displayed along with their current values. The two objects of interest are:

```
# cfgDialinUsrIndex=XX
cfgDialinUsrPppUserName=
```

If the `cfgDialinUsrPppUserName` object has no value, then that index number, which is indicated by the `cfgUserAdminIndex` object, is available for use. If a name appears after the "=", that index is taken by that user name.

 **NOTE:** When you add or remove a dial-in user with the `racadm config` subcommand, you *must* specify the index number with the `-i` option. Observe that the `cfgDialinUsrIndex` object displayed in the previous example contains a '#' character. Also, if you use the `racadm config -f racadm.cfg` command to specify any number of groups/objects to write, the index cannot be specified. A new dial-in user is added to the first available index. This behavior allows more flexibility in configuring multiple RACs with the same settings.

## Adding a Dial-In (PPP) User

To configure the dial-in user database manually. The dial-in user name and password properties have to be set for the dial-in user. You must also set the callback type. If the callback type is 1 (predefined), the callback phone number must also be supplied. Callback types are: 0=none, 1=predefined, 2=user-specified. The `<index>` must be from 1 to 16, as 16 dial-in users are supported, and they must be the same in each of the configuration commands.

```
racadm config -g cfgDialinUsers -o cfgDialinUsrPppUserName -i
<index> <username>
```

```
racadm config -g cfgDialinUsers -o cfgDialinUsrPppUserPassword -i
<index> <password>
```

```
racadm config -g cfgDialinUsers -o cfgDialinUsrCallbackType -i
<index> <callback type>
```

```
racadm config -g cfgDialinUsers -o cfgDialinUsrCallbackNumber -i
<index> <phone number>
```

The following example is sample output for configuring dial-in user number 1, with a predefined callback number.

**Example:**

```
racadm config -g cfgDialinUsers -o cfgDialinUsrPppUserName -i 1
pppaccount
```


```
racadm config -g cfgDialinUsers -o cfgDialinUsrPppUserPassword -i
1 pppaccount
```


```
racadm config -g cfgDialinUsers -o cfgDialinUsrCallbackType -i 1
0x1
```

```
racadm config -g cfgDialinUsers -o cfgDialinUsrCallbackNumber -i 1
5552222
```

## Locating the Proper Demand Dial-Out Entry Index


The RAC allows you to configure up to 16 demand dial-out entries. Before adding a demand dial-out entry, you need to know which, if any, demand dial-out entries exist. If the RAC is new, or the `racadm racresetcfg` command has been run, no demand dial-out entries exist. The `racresetcfg` subcommand resets the RAC back to the original defaults.

 **NOTICE:** Exercise caution with this command because *all* configuration parameters are reset to preinstalled settings; any previous changes are lost.

 **NOTE:** Demand dial-out entries can be added and deleted over time, so it is possible that entries on a RAC do not have the same index number as the same entry on a different RAC.

To find out if a demand dial-out entry exists, enter the following command, once for each index of 1 through 16:

```
racadm getconfig -p -g cfgDemandDial -i <index>
```


 **NOTE:** An alternative method to obtain this information would be to enter `racadm getconfig -f <myfile.cfg>`, then simply view or edit the `myfile.cfg`, which will include all RAC configuration parameters.

Six parameters or object IDs are displayed along with their current values. The two objects of interest are:

```
# cfgDemandDialIndex=XX
```

```
cfgDemandDialIpAddr=
```

If the `cfgDemandDialIpAddr` object has no value then that index number, which is indicated by the `cfgDemandDialIndex` object, is available for use. If an IP address appears after the "=", that index is already taken. Also, the index "anchor" for the demand dial-out group is not a name, but an IP address.

 **NOTE:** When you add or remove a demand dial-out entry with the `racadm config` subcommand, you *must* specify the index number with the `-i` option. Observe that the `cfgDemandDialIndex` object displayed in the previous example contains a '#' character. Also, if you use the `racadm config -f racadm.cfg` command to specify any number of groups/objects to write, that the index cannot be specified. A new demand dial-out entry is added to the first available index. This behavior allows more flexibility in configuring multiple RACs with the same settings.

## Adding a Demand Dial-Out Entry

The following example writes the five objects necessary to add a demand dial-out entry.

The demand dial-out entry remote user name and remote password properties have to be set to match the RAS user name and password. Additionally, the authentication type must be set. Authentication types are: 0=any (including clear text), 1=CHAP, 2=MSCHAP (MSCHAP is not currently supported).

**Example:**

```
racadm config -g cfgDemandDial -i <index> -o cfgDemandDialIpAddr
<destination IP address>

racadm config -g cfgDemandDial -i <index> -o
cfgDemandDialPhoneNumber <RAS phone number>

racadm config -g cfgDemandDial -i <index> -o
cfgDemandDialPppUserName <RAS username>

racadm config -g cfgDemandDial -i <index> -o
cfgDemandDialPppPassword <RAS password>

racadm config -g cfgDemandDial -i <index> -o cfgAuthType
<authentication type>
```

The following example provides sample output for configuring demand dial-out destination number 1, with CHAP authentication.

**Example:**

```
racadm config -g cfgDemandDial -i 1 -o cfgDemandDialIpAddr
143.166.154.1

racadm config -g cfgDemandDial -i 1 -o cfgDemandDialPhoneNumber
555-1212

racadm config -g cfgDemandDial -i 1 -o cfgDemandDialPppUserName
pppaccount
```

```
racadm config -g cfgDemandDial -i 1 -o cfgDemandDialPppPassword
pppaccount
```


```
racadm config -g cfgDemandDial -i 1 -o cfgAuthType 2
```

### Configuring Dial-In and Dial-Out (PPP) Properties

Two IP addresses are needed for dial-in support: a base address, and the base address incremented by one. If the base address ends with either .254 or .255, the base address will be incremented by three or two, respectively.

The base address has a default value of 0.0.0.0. When the base IP address is set to the default, the setting indicates that the base address will be obtained from `cfgDialupDialInDfltIpAddr` instead of `cfgDialupDialInBaseIpAddr`.


The default IP address has a default value of 0.0.0.0. When the default IP address is set to the default, the RAC autogenerates a unique IP address on the 10.0.0.0 network using the MAC address of the RAC network adapter.

 **NOTE:** The 10.0.0.0 network has been designated as a "private internet" and cannot be passed through gateways onto the actual Internet.

This address is unique because no two RACs autogenerate the same IP address. However, it is still possible that the autogenerated address can collide with another network node on the 10.0.0.0 network if network nodes exist other than the RAC using the 10.0.0.0 network.

When DHCP is *not* enabled for dial-in (`cfgDialupDialInDhcpEnable` is 0), the dial-in base address is obtained from:

- `cfgDialupDialInBaseIpAddr`
- OR
- `cfgDialupDialInDfltIpAddr` (if `cfgDialupDialInBaseIpAddr` has a default value of 0.0.0.0)

 **NOTE:** In this situation, the second address required for dial-in is obtained by incrementing the base address.

When DHCP is enabled for dial-in (`cfgDialupDialInDhcpEnable` is 1), the two dial-in addresses are obtained from one of the following:

- DHCP server
- The addresses assigned the last time the DHCP server responded (if a response from the DHCP server has not occurred and the prior DHCP lease has not expired)
- The `cfgDialupDialInDfltIpAddr` as subcommand a base address, and the second address obtained by incrementing the base address (if there are no previous DHCP addresses with unexpired leases do not exist and the DHCP server is not responding)

The following example specifies the necessary objects/properties to configure the RAC for PPP communication. See "RAC Property Database Group and Object Definitions" for information about the property database.

### Example:

```
racadm config -g cfgDialupNetworking -o cfgDialupDialInEnable 1
racadm config -g cfgDialupNetworking -o cfgDialupDialOutEnable 1
racadm config -g cfgDialupNetworking -o cfgDialupDialInDhcpEnable 0
racadm config -g cfgDialupNetworking -o cfgDialupDialInBaseIpAddr <IP Address>
racadm config -g cfgDialupNetworking -o cfgDialupDialInDfltIpAddr <IP Address>
racadm config -g cfgDialupNetworking -o cfgDialupDiIdleTimeout 0x3c
racadm config -g cfgDialupNetworking -o cfgDialupDiMaxConnectTime 0x0
racadm config -g cfgDialupNetworking -o cfgDialupDialType 0x1
racadm config -g cfgDialupNetworking -o cfgDialupModemInitString AT+GCI=B5
racadm config -g cfgDialupNetworking -o cfgDialupModemBaudRate 0x9600
racadm config -g cfgDialupNetworking -o cfgDialupModemPort 0x8
racadm config -g cfgDialupNetworking -o cfgDialupDialInAuthType 0x1
racadm config -g cfgDialupNetworking -o
cfgDialupDialOutModemConnectTimeout 0x1e
racadm config -g cfgRacTuning -o cfgRacTuneDialOutIdleTime 0x28
racadm config -g cfgRacTuning -o
cfgRacTuneDialOutMaxConnectTime 0x0
```

## Property Database

The RAC property database is a persistent repository of configuration information that you can modify to achieve specific behavior by the RAC firmware. When you use Server Administrator, Dell OpenManage™ IT Assistant, or the racadm utility to configure the RAC, you are making changes in the property database.

The RAC property database is structured as object groups and objects. Objects represent data types called properties (such as integer properties or boolean properties) that have property values. See "RAC Property Database Group and Object Definitions" for more information about the property database.

## Database Properties

The following sections define the user database property members and their default values.

### Telephone Number Properties

The properties that contain telephone numbers consist of a character string with the characters, as shown in Table 1-61.

**Table 1-61. Telephone Number Properties**

Character	Description
0-9	DTMF digits 0 to 9.
*	The <i>star</i> digit (tone dialing only).
#	The <i>gate</i> digit (tone dialing only).
A-D	DTMF digits A, B, C, and D. Some countries may prohibit sending of these digits during dialing.
P	Select pulse dialing: pulse-dial the numbers that follow until a "T" is encountered. Some countries prevent changing dialing modes after the first digit is dialed.
T	Select tone dialing: tone-dial the numbers that follow until a "P" is encountered. Some countries prevent changing dialing modes after the first digit is dialed.
!	Flash: the modem will go on-hook for 700ms. Not compatible with all countries.
W	Wait for dial tone: the modem will wait for dial tone before dialing the digits following "W." If the dial tone is not detected within 50 seconds, the modem will abort the rest of the sequence, return on-hook, and generate an error message.
@	Wait for silence: the modem will wait for at least 5 seconds of silence in the call progress frequency band before continuing with the next dial string parameter. If the modem does not detect these 5 seconds of silence within 50 seconds, the modem will abort the rest of the sequence, return on-hook, and generate an error message.
&	Wait for credit card dialing tone before continuing with the dial string. If the tone is not detected within 50 seconds, the modem will abort the rest of the sequence, return on-hook, and generate an error message.
,	Dial pause: the modem will pause for two seconds before dialing the digits following the ",".
()	Ignored: may be used to format the dial string.


**Table 1-61. Telephone Number Properties**

Character	Description
-	Ignored: may be used to format the dial string.

All other characters, including a blank space are invalid in the property value, and can cause unpredictable results.

### Event Filter Operation and Event Mask Properties

The RAC alert filter scans the alert enable database properties before it scans the event masks. (An event mask is a bit sequence that specifies information about the RAC- or managed-system-generated event, such as the event's origin, type, and severity).

 **NOTE:** Throughout this document, objects are always referred to by group name *and* object name, separated by a space.

The RAC alert filter operates according to the following general steps:

- It scans all of the objects in the alert enable property group.

The alert enable property group ID is `cfgUserAdmin`. The object IDs are:

- `cfgUserAdmin cfgUserAdminPageEmailEnable`
- `cfgUserAdmin cfgUserAdminPageAlphaEnable`
- `cfgUserAdmin cfgUserAdminPageNumericEnable`

If at least one of these object's property values is **TRUE**, it scans the event masks in the User table.

- It scans the `cfgTraps` `cfgTrapsEnable` object. If this object's property value is **TRUE**, it scans the event masks in the Traps table.

The following subsections describe the event masks for RAC-generated events and managed-system-generated events defined in the User table and the Trap table.

### RAC-Generated Event Mask Definitions

The `cfgUserAdmin` `cfgUserAdminAlertFilter` {`Rac`, `Sys`} `EventMask` properties are an unsigned 32-bit integer property that holds the filter information for RAC-generated events. The bit definitions in Table 1-62 apply.

**Table 1-62. RAC-Generated Event Mask Bit Definitions**

Bits	Data	Type
28–31	RAC undefined	reserved
24–27	RAC miscellaneous alerts	<Miscmask>

**Table 1-62. RAC-Generated Event Mask Bit Definitions**

Bits	Data	Type
20–23	RAC BMC communication alerts	<BMCmask>
16–19	RAC PCI AUX voltage sensor	<senMask>
12–15	RAC PCI voltage sensors	<senMask>
8–11	RAC battery voltage sensor	<senMask>
4–7	RAC external power voltage sensor	<senMask>
0–3	RAC temperature sensor	<senMask>

where RAC PCI volt sensors provide the filter mask for the following RAC sensors:

- +12 V PCI voltage sensor
- -12 V PCI voltage sensor
- +5 V PCI voltage sensor

where <senMask> has the following bit definitions:

- Bit-0: 1 = Send alert for informational events (or a return to lower severity range or normal).
- Bit-1: 1 = Send alert for warning (noncritical) events.
- Bit-2: 1 = Send alert for critical events.
- Bit-3: reserved

where <bmcMask> has the following bit definitions:

- Bit-0: 1 = Send alert for RAC communication with ESM/BMC and lost or restored communication between the RAC and ESM/BMC.
- Bit-1: 1 = Send alert for RAC lost communication with ESM/BMC.
- Bit-2: Reserved.
- Bit-3: Reserved.

where <miscMask> has the following bit definitions:

- Bit-0: 1 = Send alert for authentication failures that exceed a threshold.
- Bit-1: 1 = Send alert for battery charge low.
- Bit-2: Reserved.
- Bit-3: Reserved.

### Examples

- If you want to define alerts for the following:

- Critical voltage or temperature sensors
- Lost communication with the BMC
- Battery charge low

then the event mask property value to use is **0x02244444**. The following command sets this property value:

```
racadm -g cfgUserAdmin -o cfgUserAdminAlertFilterRacEventMask -i1 0x22444444
```

- If you want to define alerts for the following:
  - Critical events
  - System power off
  - Watchdog timer hang

then the event mask property value to use is **0x00064444**. The following command sets this property value:

```
racadm -g cfgTraps -o cfgTrapsFilterSysEventMask -i1 0x00064444
```

### ***System-Generated Alert Mask Definition***

The `cfgTraps cfgTrapsFilter {Rac, Sys} EventMask` properties are an unsigned 32-bit integer property that holds the filter information for managed-system generated events. The bit definitions in Table 1-63 apply:

**Table 1-63. System-Generated Alert Mask Bit Definitions**

<b>Bits</b>	<b>Data</b>	<b>Type</b>
28–31	System undefined	reserved
24–27	System undefined	reserved
20–23	System undefined	reserved
16–19	System status alerts	<statMask>
12–15	System miscellaneous sensor	<senMask>
8–11	System fan sensors	<senMask>
4–7	System voltage sensors	<senMask>
0–3	System temperature sensors	<senMask>

where <senMask> has the following bit definitions:

- Bit-0: 1 = Send alert for informational events (such as a return to lower severity range or normal).

- Bit-1: 1 = Send alert for warning (noncritical) events.
- Bit-2: 1 = Send alert for critical events.
- Bit-3: Reserved.

where *<statMask>* has the following bit definitions:

- Bit-0: 1 = Send alert when system transitions to a powered-on state.
- Bit-1: 1 = Send alert when system transitions to a powered-off state.
- Bit-2: 1 = Send alert when watchdog timer detects a system hang.
- Bit-3: Reserved.

### ***Alert Filter Properties***

The alert filter property group ID is `cfgUserAdmin`. The object IDs are shown in Table 1-64.

**Table 1-64. Alert Filter Property Group and Object IDs**

<b>GroupID</b>	<b>Object ID</b>	<b>Object Default Value</b>
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageNumericEnable</code>	FALSE
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageNumericNumber</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageNumericMessage</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageNumericHangupDelay</code>	0
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaEnable</code>	FALSE
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaNumber</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaMessage</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaModemConnectTimeout</code>	60 (seconds)
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaPagerId</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaProtocol</code>	8N1 (eight data bits, no parity, one stop bit)
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaBaudRate</code>	1200
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageAlphaPassword</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageEmailEnable</code>	FALSE
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageEmailAddress</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageEmailCustomMsg</code>	""
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageModemInitString</code>	ALT1M1X3S8=2E0
<code>cfgUserAdmin</code>	<code>cfgUserAdminPageModemPort</code>	0 (not used)
<code>cfgUserAdmin</code>	<code>cfgUserAdminAlertFilterRacEventMask</code>	0x777777

**Table 1-64. Alert Filter Property Group and Object IDs**

<b>GroupID</b>	<b>Object ID</b>	<b>Object Default Value</b>
cfgUserAdmin	cfgUserAdminAlertFilterSysEventMask	0x777777
cfgRemoteHosts	cfgRhostsSmtptServerIpAddr	0.0.0.0
cfgOobSnmpp	cfgOobSnmppTrapsEnable	TRUE
cfgTraps	cfgTrapsDestIpAddr	0.0.0.0
cfgTraps	cfgTrapsEnable	FALSE
cfgTraps	cfgTrapsSnmppCommunity	""
cfgTraps	cfgTrapsFilterRacEventMask	0x777777
cfgTraps	cfgTrapsFilterSysEventMask	0x777777
cfgDialupNetworking	cfgDialupModemPort	8

**Alert Test Commands**

You can test alerts using four test commands. The **racadm** command has four subcommands that test the different types of alert interfaces. These object ID sets cause the firmware to execute the subcommand with the option that indicates the test alert type to test. The test message is preset in properties for each test alert type. The four types of alerts are numeric, alphanumeric, e-mail, and trap.

The following subsection describes the command interfaces and the operation of the subcommand for each option.

***Numeric Page Test Command*****Synopsis**

```
racadm testnumpage -n -i <index>
```

```
racadm testnumpage -n -u <username>
```

**Alert Data Definitions**

The contents of the **cfgUserAdmin** **cfgUserAdminPageNumericMessage** string are sent following dialing of the **cfgUserAdmin** **cfgUserAdminPageNumericNumber** string. Several characters have special meanings. See "Telephone Number Properties."

The special character "S" causes the RAC to fill in the severity of the alert:

- 0 = Informational/normal
- 1 = Warning (noncritical)
- 2 = Critical

The special character "E" causes the RAC to fill in the event type:

- 001 = Test page (informational/normal)

- 002 = RAC authentication errors (warning)
- 003 = Lost communication with ESM error (critical)
- 004 = Communication with ESM restored (informational/normal)
- 005 = Detected system power-off (critical)
- 006 = Detected system power-on (informational/normal)
- 007 = RAC detected that the system watchdog expired (system hang) (critical)
- 008 = RAC battery charge low (minor) (DRAC III only)
- 009 = Hardware log 80% full
- 010 = Hardware log 90% full
- 011 = Hardware log 100% full
- 101 = RAC temperature sensor (informational/warning/critical) (DRAC III only)
- 102 = RAC voltage sensor (informational/warning/critical)
- 201 = System temperature sensor (informational/warning/critical)
- 202 = System voltage sensor (informational/warning/critical)
- 203 = System fan sensor (informational/warning/critical)
- 204 = Other system sensor (informational/warning/critical)

The special character "N" causes the RAC to fill in the sensor number for a system event.

For RAC events, the sensors are numbered as follows (DRAC III only):

- 1 = RAC temp sensor
- 2 = RAC battery voltage
- 3 = RAC external power supply voltage
- 4 = RAC +12 V PCI voltage
- 5 = RAC -12 V PCI voltage
- 6 = RAC +5 V PCI voltage
- 7 = RAC AUX PCI voltage

The special character "I" causes the RAC to fill in the IP address of the network adapter in dotted-decimal notation, but without the dots.

"999" is prepended to the `cfgUserAdmin cfgUserAdminPageNumericMessage` object for test pages. The concatenated string is then interpreted as previously described.

### ***Alpha Numeric Page Test Command***

#### **Synopsis**

```
racadm testalphapage -a -i <index>
```

```
racadm testalphapage -a -u <username>
```

### Alert Data Definitions

The alphanumeric alert contains the following information: message (including test message, if a paging test), event description, date, time, severity, sensor reading, system ID, model, BIOS version, asset tag, service tag, managed system name, operating system type, and operating system name. The following is an example test alphanumeric page:

```
Subject: Alert from Dell Remote Access Card: 10.35.10.108
Message: TEST PAGE
Hostname:53ewy
Event: Email paging test to user 1
Date: 06-jun-2001
Time: 00:01:37
Severity: Info/Normal
```

### ***E-Mail Test Command***

#### Synopsis

```
racadm testemail -e -i <index>
racadm testemail -e -u <username>
```

### Alert Data Definitions

The e-mail alert contains the following information: message (including test message, if a paging test), event description, date, time, severity, system ID, model, BIOS version, asset tag, service tag, managed system name, operating system name, and ESM version. The following is an example test e-mail (fields shown are examples only and may not reflect actual observed output for your environment):

```
Subject: Alert from Dell Remote Access Card: 10.35.10.108
Message: TEST PAGE
Event: Email paging test to user 1
Date: 06-jun-2001
Time: 00:01:37
Severity: Info/Normal
System ID: Bbn
Model: Dell PowerEdge 7150
BIOS version: A00
```

```
Asset tag: 181676
Service tag: 6X713
Hostname: P2-750-08
OS Name: Linux 7.1 for the Itanium Processor
ESM Version: 1.3
```

### ***Trap Test Command***

#### **Synopsis**

```
racadm testtrap -t -i <trap index>
```

#### **Alert Data Definitions**

The "alertMessage" string (up to 1 KB) provides the specific information describing the cause and specific source of the event, which includes:


- Sensor identification: entity/IPMBslaveAddress
- Sensor number
- Sensor ID string (if possible)
- Current reading and range (normal/warning/critical)
- Threshold values: Minimum, maximum, normal

For more information, see the *Server Administrator SNMP Reference Guide*.

### **Configuration Using the racadm Utility and the racadm.cfg File**

One of the major features of the racadm utility is the ability to configure the RAC using a configuration file. The racadm utility parses the RAC configuration file, called **racadm.cfg**, and then sends individual configuration requests to one or more RACs.


This method may be used to configure multiple RAC database properties. You must first run the racadm utility to query a configured RAC for its database properties, which are accessed using their object group IDs and object IDs. The racadm utility creates the **racadm.cfg** file from the retrieved information. You can then configure other cards with the same database information by exporting this file out to the other RACs.

 **NOTE:** Some configuration files contain unique RAC information (such as static IP address) that must be modified before configuring other cards.

The following are the general steps for this procedure:

- 1** Get the configuration from the RAC that contains the appropriate configuration.
- 2** Modify the configuration (optional).
- 3** Push the configuration to a target RAC.
- 4** Reset the target RAC.

The `getconfig -f racadm.cfg` subcommand requests the configuration of the RAC and generates a `racadm.cfg` file (you can choose any name for this file).

 **NOTE:** The generated `.cfg` file does not contain user passwords.

Other options for `getconfig` enable you to perform such actions as:


- Displaying all configuration properties in a group (specified by group name and index)
- Displaying all configuration properties for a user by user name
- Including password (other than user password) information in the generated `.cfg` file

The `config` subcommand loads the information into other RACs. Other options for `config` enable you to perform such actions as:

- Removing passwords in the `racadm.cfg` file used to configure the card
- Synchronizing the user and password database with Server Administrator

The initial configuration file, `racadm.cfg`, is named by the user. In the following example, the configuration file is named `myfile.cfg`. To obtain this file, type the following command at the command prompt:

```
racadm getconfig -f myfile.cfg
```

 **NOTICE:** It is recommended that you edit this file with a simple text editor; the `racadm` utility uses an ASCII text parser, and any formatting confuses the parser and might corrupt the `racadm` database.

The following example describes the IP address of the RAC. Remove all unnecessary `<variable>=value` entries. In this situation, only the actual variable group's label with "[" and "]" will remain along with the two `<variable>=value` entries pertaining to the IP address change.

The file contents are as follows:

```
#
# Object Group "cfgLanNetworking"
#
[cfgLanNetworking]
cfgNicIpAddress=10.35.10.110
cfgNicGateway=10.35.10.1
```

This file will be updated as follows:

```
#
# Object Group "cfgLanNetworking"
#
[cfgLanNetworking]
cfgNicIpAddress=10.35.9.143
```

```
# comment, the rest of this line is ignored
cfgNicGateway=10.35.9.1
```

The command `racadm config -f myfile.cfg` parses this file and identifies any errors by line number. A correct file will update the proper entries. You may use the same `getconfig` command used in the previous example to confirm the update.

You can use this file to download company-wide changes or to configure new systems over the network.

## RAC Configuration File

The RAC configuration file `<filename>.cfg` is used with the `racadm config -f <filename>.cfg` command. The configuration file is a simple text file that allows the user to build a configuration file (similar to an `.ini` file) and configure the RAC from this file. You may use any file name, and the file does not require a `.cfg` ending (although it is referred to by that designation in this subsection). The `.cfg` file can be:

- Created
- Obtained from a `racadm getconfig -f <filename>.cfg` command
- Obtained from a `racadm getconfig -f <filename>.cfg` command, and then edited



**NOTE:** See "config/getconfig" for information about the `getconfig` command.

The `.cfg` file is first parsed to verify that valid group and object names are present and that some simple syntax rules are being followed. Errors are flagged with the line number in which the error was detected, and a simple message explains the problem. The entire file is parsed for correctness, and all errors are displayed. Writes are not performed to the RAC if an error is found in the `.cfg` file. The user must correct *all* errors before any configuration can take place. The `-c` option may be used in the `config` subcommand, which verifies syntax only and does *not* perform writes to the RAC.

Remember the following important points:

- If the parser encounters an indexed group, it is the value of the anchored object that differentiates the various indexes.

The parser reads in all of the indexes from the RAC for that group. Any objects within that group are simple modifications at configuration time. If a modified object represents a new index, the index is created on the RAC during configuration.

- The user cannot specify a desired index in a `.cfg` file.

Indexes may be created and deleted, so over time the group may become fragmented with used and unused indexes. If an index is present, it is modified. If an index is not present, the first available index is used. This method allows flexibility when adding indexed entries, where the user does not need to make exact index matches between all of the RACs being managed; new users are added to the first available index. A `.cfg` file that parses and runs

correctly on one RAC may not run correctly on another if all of indexes are full and a new user was to be added.

- Use the **racresetcfg** subcommand to keep all RACs the same.

To keep all RACs the same, use the **racresetcfg** subcommand to reset the RAC to original defaults, and then run the **racadm config -f <filename>.cfg** command. Ensure that the **.cfg** file has all the desired objects, users, indexes, and other parameters.



**NOTICE:** Using the **racresetcfg** subcommand resets the database and the RAC network adapter settings to the original default settings and removes all users and user configurations. While the root user is available, other users' settings are also reset to the default.

## Parsing Rules

- All lines that start with '#' are treated as comments.

A comment line *must* start in column one. A '#' character in any other column is treated as a # character. (Some modem parameters may have # characters as part of their string. An escape character is not required. The user may want to generate a **.cfg** from a **racadm getconfig -f <filename>.cfg** command, and then perform a **racadm config -f <filename>.cfg** command to a different RAC, without adding escape characters).

**Example:**

```
#  
# This would be a comment  
[cfgUserAdmin]  
cfgUserAdminPageModemInitString=<Modem init # not a comment>
```

- All group entries must be surrounded by "[" and "]" characters.

The starting "[" character denoting a group name *must* start in column one. This group name *must* be specified before any of the objects in that group. Objects that do not have a group name associated with them generate an error. The configuration data is organized into groups as defined in "RAC Property Database Group and Object Definitions."

The following example is of a group name, object, and the object's property value.

**Example:**

```
[cfgLanNetworking]  
cfgNicIpAddress=143.154.133.121
```

- All parameters are specified as "object=value" pairs without any white space between the object, =, or value.

White spaces after the value are ignored. A white space inside a value string is left unmodified. Any character to the right of the '=' is taken as is (for example, a second '=', or a '#', '[', ']', and so forth). All of these characters are valid modem chat script characters.

See the example in the previous bullet item.

- An indexed object entry is ignored by the .cfg parser.

The user *cannot* specify which index is used. If the index already exists, it is used, or else the new entry is created in the first available index for that group. The **racadm getconfig -f <filename>.cfg** command places a comment in front of index objects, which allows the user to see which comments are being used.



**NOTE:** The user may create an indexed group manually using the following command:

```
racadm config -g <groupName> -o <anchored object> -i <index 1-16> <unique anchor name>
```

- The line for an indexed group *cannot* be deleted from a .cfg file.

The user must remove an indexed object manually using the following command:

```
racadm config -g <groupName> -o <objectName> -i <index 1-16> ""
```



**NOTE:** A NULL string (two " characters [""]) directs the RAC to delete the index for the specified group.

To view the contents of an indexed group, use the following command:

```
racadm getconfig -g <groupName> -i <index 1-16>
```


- For indexed groups the object anchor *must* be the first object after the "[" ]" pair. The following are examples of the current indexed groups:

```
[cfgUserAdmin]
cfgUserAdminUserName=<USER_NAME>
"

[cfgTraps]
cfgTrapsDestIpAddress=<IP_ADDRESS>
"
"

[cfgDemandDial]
cfgDemandDialIpAddress=<IP_ADDRESS>
"
```

```
[cfgDialinUsers]
cfgDialinUsrPppUserName=<USER_NAME>
,
,
```

 **NOTE:** Enter the `racadm getconfig -f <myexample> .cfg` command. This command builds a `.cfg` file of the current RAC configuration. This file can be used as an example and as a starting point for your unique `.cfg` file.



# RAC Property Database Group and Object Definitions

The RAC property database contains the configuration information for the RAC. Data is organized by associated object, and objects are organized by object group. The IDs for the groups and objects that the property database supports are listed in this section.

The group and object IDs provide input to the racadm utility to set and retrieve RAC data. The following definitions specify object data and indicate whether the object is readable, writable, or both.

The following subsections describe the groups and their member objects.



**NOTICE:** Object and group IDs in this section are system specific. Some may not be applicable to your system.

## idRacInfo

This group contains display parameters to provide information about the specifics of the RAC being queried.

One instance of the group is allowed. The following subsections describe the objects in this group.

### idRacType (Read Only)

#### Legal Values

0=invalid, 1=other, 2=unknown, 3=DRAC III, 4=all other RACs

#### Default

0x3 or 0x4

#### Description

Identifies the type of RAC.

### idRacProductInfo (Read Only)

#### Legal Values

String of up to 63 ASCII characters.

**Default**

RAC

**Description**

Using a text string, identifies what the product is.

**idRacDescriptionInfo (Read Only)**

**Legal Values**

String of up to 255 ASCII characters.

**Default**

RAC

**Description**

A text description of the RAC type.

**idRacVersionInfo (Read Only)**

**Legal Values**

String of up to 63 ASCII characters.

**Default**

RAC Firmware Version x.x

**Description**

A string containing the current firmware version of the product, where x is the current revision.

**idRacName (Read/Write)**

**Legal Values**

String of up to 15 ASCII characters.

**Default**

RAC

**Description**

A user assigned name to identify this controller.

## **idPcmcialInfo (Read Only)**

### **Legal Values**

String of up to 63 ASCII characters

### **Default**

Null string

### **Description**

Text description of the PCMCIA modem card.

## **idRacMisc (Read/Write)**

### **Legal Values**

String of up to 63 ASCII characters.

### **Default**

Null string

### **Description**

Generic property undefined at this release.

## **cfgMonitor**

This group contains parameters to configure the monitoring of the optional external AC power adapter and the onboard battery.

One instance of the group is allowed. The following subsections describe the objects in this group.

### **cfgMonitorExtPwrScrRule (Read/Write)**

#### **Legal Values**

0x0, 0x1, or 0x2

#### **Default**

0x2

#### **Description**

0=Disable, do not monitor the external power source.

1=Always monitor the external power source.

2=Only monitor the external power source if it is connected.

### **cfgMonitorBatteryRule (Read/Write)**

#### **Legal Values**

0x1

#### **Default**

0x1

#### **Description**

The battery is not optional at the first release; 1= always monitor the onboard battery.

## **cfgLanNetworking**

This group contains parameters to configure the RAC network adapter interface.

One instance of the group is allowed. All objects in this group require a RAC reset before they become active. The following subsections describe the objects in this group.

### **cfgNicEnable (Read/Write)**

#### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).


#### **Default**

1

#### **Description**

0=Disable.

1=Enable the RAC network adapter.

 **NOTE:** This object requires a RAC reset before it becomes active.

### **cfgNicIpAddress (Read/Write)**

#### **Legal Values**

A string of "." separated numeric fields containing the static IP address.

#### **Default**

192.168.20.1

**Description**

The IP address of the RAC network adapter.

**cfgNicNetmask (Read/Write)****Legal Values**

A string of "." separated numeric fields containing the static network mask.

**Default**

255.255.255.0

**Description**

The network mask used by the RAC network adapter.



**NOTE:** This object requires a RAC reset before it becomes active.

**cfgNicGateway (Read/Write)****Legal Values**

A string of "." separated numeric fields containing the static gateway.

**Default**

192.168.20.1

**Description**

The gateway used by the RAC network adapter.

**cfgNicUseDhcp (Read/Write)****Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE)

**Default**

0

**Description**

0=Use the static RAC network adapter parameters described above.

1=Use DHCP and obtain the necessary parameters from the DHCP server for the RAC network adapter.

## **cfgCurrentLanNetworking**

This group contains parameters that are currently in use by the RAC network adapter interface. One instance of the group is allowed. The following subsections describe the objects in this group.

### **cfgNicCurrentIpAddress (Read Only)**

#### **Legal Values**

A string of "." separated numeric fields containing the IP address

#### **Default**

None

#### **Description**

The current IP address of the RAC network adapter.

### **cfgNicCurrentNetmask (Read Only)**

#### **Legal Values**

A string of "." separated numeric fields containing the network mask.

#### **Default**

None

#### **Description**

The current network mask used by the RAC network adapter.

### **cfgNicCurrentGateway (Read Only)**

#### **Legal Values**

A string of "." separated numeric fields containing the gateway address.

#### **Default**

None

#### **Description**

The current gateway used by the RAC network adapter.

## **cfgNicCurrentDhcpWasUsed (Read Only)**

### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

### **Default**

None

### **Description**


Indicates whether or not DHCP was used to configure the network adapter.

0 = IP address is static

1 = IP address was obtained from a DHCP server.

## **cfgDialupNetworking**

This group contains parameters to configure the capabilities of the DRAC III dial-in and dial-out (demand dial).

 **NOTE:** This object group is for the DRAC III controller only.

One instance of the group is allowed. All objects in this group require a RAC reset before they become active. The following subsections describe the objects in this group.

### **cfgDialupDialInEnable (Read/Write)**

#### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

#### **Default**

1

#### **Description**

0=Enable, 1=Disable the dial in capabilities.

### **cfgDialupDialOutEnable (Read/Write)**

#### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

#### **Default**

1

**Description**

0=Enable, 1=Disable the dial out capabilities.

**cfgDialupDialInDhcpEnable (Read/Write)****Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

**Default**

0

**Description**

0=Enable, 1=Disable. Get the IP address from the DHCP server to becomes the PPP IP address(es). Depending on the network adapter configuration either 1 or 2 DHCP addresses are used. (the RAC is the client.)

**cfgDialupDialInBaseIpAddr (Read/Write)****Legal Values**

A string of "." separated numeric fields containing the IP address.

**Default**

0.0.0.0

**Description**

Static IP base address for PPP. When static addresses are enabled two IP addresses are used. The first is the base, and the second is base+1. When the base IP address is 0.0.0.0, **cfgDialupDialInDfltIpAddr** is used instead.

For more information, see the *Dell Remote Access Controller Installation and Setup Guide*.

**cfgDialupDialInDfltIpAddr (Read/Write)****Legal Values**

A string of "." separated numeric fields containing the IP address.

**Default**

Autogenerated

**Description**

Statically configure the IP base address for PPP. Used when either `cfgDialupDialInBaseIpAddr` is 0, or DHCP fails to obtain IP addresses from the DHCP server (only when DHCP is enabled using `cfgDialupDialInDhcpEnable`). When set to 0.0.0.0, a unique address on the 10.0.0.0 network is autogenerated.

You can use this command to avoid the issue described in "Accessing the DRAC III From a Dial-In Client Running Red Hat Linux (With PPP-2.3 or Later)" in the *Remote Access Controller Installation and Setup Guide*.

**cfgDialupDialIdleTimeout (Read/Write)****Default**

0x3C (60 seconds)

**Description**

0=infinite, otherwise when line is idle for this many seconds, the line is disconnected. (Enter hexadecimal values.)

**cfgDialupDiMaxConnectTime (Read/Write)****Default**

0x0

**Description**

0=Infinite, otherwise maximum time in seconds that the link will stay connected before the connection is terminated. (Enter hexadecimal values.)

**cfgDialupDialType (Read/Write)****Default**

0x1

**Description**

0x0=pulse, 0x1=tone (Enter hexadecimal values.)

**cfgDialupModemInitString (Read/Write)****Legal Values**

String of up to 63 ASCII characters meaningful to the modem.

**Default**

ATL1M1X3E0S0=2

**Description**

A command string passed to the modem to initialize it.

**cfgDialupModemBaudRate (Read/Write)****Default**

0x9600 or 38400

**Description**

Baud rate of the modem. (Enter hexadecimal values.)

**cfgDialupDialInAuthType (Read/Write)****Default**

0x1

**Description**

0=PAP protocol, 1=CHAP, 2=MS CHAP (MS CHAP is not supported). (Enter hexadecimal values.)

**cfgDialupDialOutModemConnectTimeout (Read/Write)****Default**

0x1E or 30

**Description**

0=infinite, otherwise the maximum time in seconds that the link will stay connected before the connection is terminated. (Enter hexadecimal values.)

**cfgRemoteHosts**

The group contains parameters to configure various firmware update loading, remote floppy boot, IP addresses, enables, and so on.

One instance of the group is allowed. The following subsections describe the objects in this group.

### **cfgRhostsSmtEmailEnable (Read/Write)**

#### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

#### **Default**

1

#### **Description**

0=disable, 1=enable the SMTP protocol to send e-mail alerts. This object requires a RAC reset before it becomes active.

### **cfgRhostsFloppyTftpEnable (Read/Write)**

#### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

#### **Default**

0

#### **Description**

0=Disable, 1=Enable remote floppy boot. This object requires a RAC reset before it becomes active.

### **cfgRhostsFwUpdateTftpEnable (Read/Write)**

#### **Legal Values**

Boolean either 1 or 0 (TRUE or FALSE).

#### **Default**

1

#### **Description**

0=Disable, 1=Enable loading the firmware update file through TFTP. This object requires a RAC reset before it becomes active.

### **cfgRhostsSmtServerIpAddr (Read/Write)**

#### **Legal Values**

A string of "." separated numeric fields containing the IP address.

**Default**

127.0.0.1

**Description**

The IP address of the server used in e-mail alerts. This object requires a RAC reset before it becomes active.

**cfgRhostsFloppyTftpIpAddr (Read/Write)****Legal Values**

A string of "." separated numeric fields containing the IP address.

**Default**

192.168.0.3

**Description**

The IP address of the TFTP server where the boot image is located.

**cfgRhostsFloppyTftpPath (Read/Write)****Legal Values**

String of up to 255 ASCII characters that designate a valid path name.

**Default**

BOOTIMG.BIN

**Description**

The path name pointing to the remote floppy boot image. If this is a file name only, the path needs to be specified in the TFTP server, or the entire path can be specified here.



**NOTE:** The server may still require you to specify the drive, such as C drive.

**cfgRhostsFwUpdateIpAddr (Read/Write)****Legal Values**

A string of "." separated numeric fields containing the IP address.

**Default**

192.168.0.4

**Description**

The address of the TFTP server where the firmware update image is located.

**cfgRhostsFwUpdatePath (Read/Write)****Legal Values**

String of up to 255 ASCII characters that designate a valid path name.

**Default**

FIRMIMG.BIN

**Description**

The path name pointing to the firmware update binary file. If this is a file name only, then the path needs to be specified in the TFTP server. Otherwise, the entire path can be specified here.



**NOTE:** The server may still require you to specify the drive, such as C drive.

## cfgUserAdmin

This group contains parameters to allow configuration of users that are allowed access to the RAC.

Sixteen instances of the group are allowed, which corresponds to a user for each index. The following subsections describe the objects in this group.

**cfgUserAdminUserName (Read/Write)****Legal Values**

A string of up to 19 ASCII characters.

**Default**

None

**Description**

The name of the user for this index. The user index is created by writing a string into this name field if the index is empty. Writing a "" string deletes the user at that index. You cannot change the name. You must delete and then recreate.



**NOTE:** This command is the "anchor" for this indexed group.

**cfgUserAdminPassword (Write Only)****Legal Values**

A string of up to 255 ASCII characters

**Default**

None

**Description**

The password for this user. The user passwords are encrypted and cannot be seen or displayed after this property is written.

**cfgUserAdminAlertFilterRacEventMask (Read/Write)****Legal Values**

See "RAC-Generated Event Mask Definitions."

**Default**

0x777777

**Description**

See "RAC-Generated Event Mask Definitions." (Enter hexadecimal values.)

**cfgUserAdminAlertFilterSysEventMask (Read/Write)****Legal Values**

See "System-Generated Alert Mask Definition."


**Default**

0x777777

**Description**

See "System-Generated Alert Mask Definition." (Enter hexadecimal values.)

**cfgUserAdminPageNumericEnable (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).


**Default**

0

**Description**

0=Disable, 1=Enable numeric paging on a per user basis.

**cfgUserAdminPageNumericNumber (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

A string of up to 95 ASCII characters.


**Default**

""

**Description**

See "Telephone Number Properties" for valid phone number characters.

**cfgUserAdminPageNumericMessage (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**


A string of up to 32 ASCII characters.

**Default**


SE

**Description**

A string of digits consisting of the following: 0-9, SENI. In the numeric message, S will be replaced by the decimal value of the message severity. The E is replaced by the three-decimal digit event number. The N is replaced by the two-decimal digit sensor number. The I is replaced by the IP address of the RAC (dotted-decimal notation with the dots deleted).

 **NOTICE:** It is possible to overflow the number of digits that a numeric pager can handle. Therefore, you must first determine the maximum number of digits handled by the paging service and equipment in use.

**cfgUserAdminPageNumericHangupDelay (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

Integer from 0 to any positive number

**Default**

0x0

**Description**

Number of seconds to extend the time duration of a numeric page. (Enter hexadecimal values.)

**cfgUserAdminPageAlphaEnable (Read/Write)**

**NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

Boolean either 1 or 0 (TRUE or FALSE).

**Default**

0

**Description**

0=Disable, 1=Enable alpha numeric paging on a per user basis.

**cfgUserAdminPageAlphaNumber (Read/Write)**

**NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

A string of up to 95 ASCII characters.

**Default**

""

**Description**

See "Telephone Number Properties" for valid telephone number characters.

**cfgUserAdminPageAlphaProtocol (Read/Write)**

**NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

8N1 or 7E0


**Default**

8N1

**Description**

Number of data bits, N=no parity, E=even parity, number of stop bits.

**cfgUserAdminPageAlphaBaudRate (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

Integer from 0 to 115200.


**Default**

0x4B0 or 1200

**Description**

Baud rate for the modem. (Enter hexadecimal values.)

**cfgUserAdminPageAlphaCustomMsg (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

A string of up to 31 ASCII characters.


**Default**

""

**Description**

User defined message to be sent on an alpha page.

**cfgUserAdminPageAlphaModemConnectTimeout (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

Integer from 0 to any positive number.


**Default**

0x3C or 60

**Description**

0=Infinite. The time in seconds that the remote system has to answer the telephone. (Enter hexadecimal values.)

**cfgUserAdminPageAlphaPagerId (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

A string of up to 31 ASCII characters.


**Default**

""

**Description**

Assigned by the paging service.

**cfgUserAdminPageAlphaPassword (Read/Write)**

 **NOTE:** This object property is for the DRAC III controller only.

**Legal Values**

A string of up to 31 ASCII characters.

**Default**

""

**Description**

Assigned by the paging service.

**cfgUserAdminEmailEnable (Read/Write)****Legal Values**

Boolean either 1 or 0 (TRUE or FALSE).

**Default**

0

**Description**

0=Disable, 1=Enable e-mail alerting on a per user basis.

## **cfgUserAdminEmailAddress (Read/Write)**

### **Legal Values**

A string of up to 63 ASCII characters.

### **Default**

""

### **Description**

Standard e-mail address, such as john\_doe@mycompany.com.

## **cfgUserAdminEmailCustomMsg (Read/Write)**

### **Legal Values**

A string of up to 31 ASCII characters.

### **Default**

""

### **Description**

User-defined message to be sent on a e-mail alert.

## **cfgUserAdminPageModemInitString (Read/Write)**



**NOTE:** This object property is for the DRAC III controller only.

### **Legal Values**

A string of up to 31 ASCII characters.

### **Default**

AT+GCI=B5

### **Description**

Characters that make up a legal modem initialization string.

## **cfgTraps**

This group contains parameters to configure the delivery of SNMP traps.

Sixteen instances of this group are allowed, which represent sixteen unique trap destinations. The following subsections describe the objects in this group.

**cfgTrapsDestIpAddr (Read/Write)****Legal Values**


A string of "." separated numeric fields containing the IP.

**Default**

""

**Description**

IP address of an SNMP trap daemon.

 **NOTE:** This object is the "anchor" for this indexed group.

**cfgTrapsEnable (Read/Write)****Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

**Default**

1

**Description**

0=Disabled, 1=Enabled for this indexed entry.

**cfgTrapsSnmpCommunity (Read/Write)****Legal Values**

A string of up to 31 ASCII characters.

**Default**

""

**Description**

A SNMP community name.

**cfgTrapsFilterRacEventMask (Read/Write)****Legal Values**

See "RAC-Generated Event Mask Definitions."

**Default**

0x777777

**Description**

See "RAC-Generated Event Mask Definitions." (Enter hexadecimal values.)

**cfgTrapsFilterSysEventMask (Read/Write)****Legal Values**

See "RAC-Generated Event Mask Definitions."

**Default**


0x777777

**Description**

See "RAC-Generated Event Mask Definitions." (Enter hexadecimal values.)

## cfgDemandDial

This group contains parameters to configure dial-out capabilities of the DRAC III controller.

 **NOTE:** This object group is for the DRAC III controller only.

Sixteen instances of the group are allowed, which represent sixteen dial out destinations. The telephone numbers need not be unique among each index. However, the IP address must be unique; which means multiple IP addresses may use the same telephone number. All objects in this group require a RAC reset before they become active. The following subsections describe the objects in this group.

**cfgDemandDialIpAddr (Read/Write)****Legal Values**


A string of "." separated numeric fields containing the IP.

**Default**

""

**Description**

This address is the destination IP address. This IP address may not be the IP address of the system that responds to the telephone number. However, the system that responds to the telephone number must be able to route to this IP address.

 **NOTE:** This object is the "anchor" for this indexed group.

**cfgDemandDialPhoneNumber (Read/Write)****Legal Values**

A string of up to 95 ASCII characters.

**Default**

""

**Description**

The telephone number of a system that is able to route packets to the IP address set in `cfgDemandDialIpAddress`. See "Telephone Number Properties" for valid telephone number characters.

**cfgDemandDialPppUserName (Read/Write)****Legal Values**

A string of up to 31 ASCII characters.

**Default**

""

**Description**

User name for PPP authentication.

**cfgDemandDialPppPassword (Read/Write)****Legal Values**

A string of up to 31 ASCII characters.

**Default**

""

**Description**

Password for PPP authentication.

**cfgAuthType (Read/Write)****Legal Values**

0, 1, 2

**Default**


0x0

**Description**

PPP authentication type; 0=Any (including clear text) 1=CHAP, 2=MS CHAP (MS CHAP not supported). (Enter hexadecimal values.)

## cfgDialinUsers

This group contains parameters to configure the dial in PPP authentication capability of the DRAC III.

 **NOTE:** This object group is for the DRAC III controller only.

Sixteen instances of the group are allowed, where each index represents a single PPP dial-in user. All objects in this group require a RAC reset to become active. The following subsections describe the objects in this group.

**cfgDialinUsrPppUserName (Read/Write)****Legal Values**


A string of up to 15 ASCII characters.

**Default**

""

**Description**

User name for PPP authentication.

 **NOTE:** This object is the "anchor" for this indexed group.

**cfgDialinUsrPppUserPassword (Read/Write)****Legal Values**

A string of up to 15 ASCII characters.

**Default**

""

**Description**

Password for PPP authentication.

**cfgDialinUsrCallbackType (Read/Write)****Default**

0x0

**Description**

0=None, 1=Admin preset, 2=User-specified. Used by the callback control protocol (CBCP). (Enter hexadecimal values.)

**cfgDialinUsrCallbackNumber (Read/Write)****Legal Values**

A string of up to 95 ASCII characters.

**Default**

""

**Description**

Administrative preset telephone number for CBCP if a 1 is selected in **cfgDialinUsrCallbackType**. See "Telephone Number Properties" for valid phone number characters.

**cfgSessionManagement**

This group contains parameters to configure the number of sessions that can connect to the RAC.

One instance of the group is allowed. All objects in this group require a RAC reset before they become active. The following subsections describe the objects in this group.

**cfgSsnMgtMaxConRedirSessions (Read/Write)****Legal Values**

0x1, 0x2, 0x3, 0x4

**Default**

0x4

**Description**

The maximum number of simultaneous console redirect sessions that are allowed at one time from the RAC Web-based remote access interface. (Enter hexadecimal values.)

## **cfgSsnMgtMaxSessions (Read/Write)**

### **Legal Values**

0x1 through 0x10

### **Default**

0x10 or 16

### **Description**

The maximum number of simultaneous sessions that are allowed at one time from the RAC Web-based remote access interface. (Enter hexadecimal values.)

## **cfgSsnMgtMaxSessionsPerUser (Read/Write)**

### **Legal Values**

0x1 through 0x10

### **Default**

0x10 or 16

### **Description**

The maximum number of simultaneous sessions allowed per user. (Enter hexadecimal values.)

## **cfgSerial**

This group contains configuration parameters for the system external serial port or DRAC III VT-100 serial connector.

One instance of the group is allowed. The following subsections describe the objects in this group.

### **cfgSerialBaudRate (Read/Write)**



**NOTE:** For best results redirecting BIOS System Setup, Dell recommends using 115200. For the DRAC III controller, Dell recommends using 9600.

### **Legal Values**

9600, 28800, 57600, 115200

### **Default**

115200 for all RACs except the DRAC III; 9600 for DRAC III

**Description**

Sets the baud rate on the external serial port. (Enter decimal values).

**cfgSerialConsoleEnable (Read/Write)****Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

**Default**

0 for all RACs except the DRAC III controller; 1 for the DRAC III controller

**Description**

0=Disabled, 1=Enabled. Enables the serial port and terminal interface. This object requires a RAC reset before it becomes active.

**cfgSerialConsoleQuitKey (Read/Write)****Legal Values**

A string of 3 or less characters.

**Default**

The <CR><~><. > key combination except for the DRAC III controller; use <F3> for the DRAC III controller

The <CR> key represents a carriage return; press <Enter> as a substitute for <CR>.

**Description**

This key sequence terminates text console redirection when using VT-100.

**cfgSerialConsoleIdleTimeout (Read/Write)****Legal Values**

Integer from 1 to any positive number. Enter hexadecimal values.

**Default**

0x12c

**Description**

The maximum number of time (in seconds) of line idle time before the line is disconnected. (Enter hexadecimal values.)

## **cfgSerialConsoleShellType (Read/Write)**

### **Default**

Default 2 except for the DRAC III; 1 for the DRAC III

### **Legal Values**

1 = DRAC III style VT100 block screen interface, has limited command function compared to type 2.

2 = UNIX<sup>®</sup>-style commandline data stream interface.

3 = Dial-in to Unix-style commandline data stream interface. An external modem must be attached to the external serial port (for the DRAC III this port is the serial cable).

4 = PPP dial-in through an external modem. An external modem must be attached to the external serial port (on DRAC III this is the serial cable).

## **cfgSerialConsoleNoAuth (Read/Write)**

### **Description**

Allows user to disable authentication on serial shell.

### **Default**

0

### **Legal Values**

0 – Login prompt is **Enabled** on the serial shell.

1 – Login prompt is **Disabled** on serial shell.

## **cfgSerialConsoleCommand (Read/Write)**

### **Description**

The serial command runs after login at the start of a session and allows you to set up a command such as `connect com2` that autoruns when a session begins.

### **Default**

Empty string (no command).

### **Example**

```
connect com2
```

**cfgSerialTelnetEnable (Read/Write)****Description**

Enables/disables telnet console

**Default**

0=telnet disabled

**Legal Values**

0=disabled, 1=enabled

**cfgSerialConsoleColor (Read/Write)****Description**

Enables/disables the serial text console color when connecting using the **connect video** command.

**Default**

Default is 1 for all RACs except the DRAC III; 0 for the DRAC III

**Legal Values**

0=monochrome, 1=color

**cfgSerialConsoleAnsi (Read/Write)**

Enables/disables the serial text console ANSI graphics mode when connecting using the **connect video** command.

**Default**

1=ANSI enabled

**Legal Values**

0=VT100 graphics mode, 1=ANSI graphics mode

**cfgSerialTelnetColor (Read/Write)****Description**

Enables/disables the telnet text console color when connecting using the **connect video** command.

**Default**

1

**Legal Values**

0=monochrome, 1=color

**cfgSerialTelnetAnsi (Read/Write)****Description**

Enables/disables the telnet text console ANSI graphics mode when connecting using the **connect video** command.

**Default**

1

**Legal Values**

0 =VT100 graphics mode, 1=ANSI graphics mode

**cfgSerialCom2RedirEnable (Read/Write)****Description**

Enables/disables the console through the COM 2 port (not available on DRAC III)

**Default**

1=COM 2 (console enabled)

**Legal Values**

0=Disabled, 1=Enabled

**cfgNetTuning**

The group contains parameters to tune the RAC network configuration.

One instance of the group is allowed. All objects in this group require a RAC reset before they become active. The following subsections describe the objects in this group.

**cfgNetTuningNicMtu (Read/Write)****Legal Values**

Integer from 576 to 1500.

**Default**

0x5dc (1500).

**Description**

The size in bytes of the maximum transmission unit used by the RAC network adapter. (Enter hexadecimal values.)

**cfgNetTuningIpTtl (Read/Write)****Legal Values**

Integer from 1 to 255.

**Default**

0x40 (64)

**Description**

The maximum IP packet time to live in seconds. (Enter hexadecimal values.)

**cfgNetTuningIpReassTtl (Read/Write)****Legal Values**

Integer from 60 to 240.

**Default**

0x3c (60)

**Description**

The maximum IP packet fragment reassembly time in 1/2 second units. (Enter hexadecimal values.)

**cfgNetTuningTcpSrttBase (Read/Write)****Legal Values**

Integer from 0 to 256.

**Default**

0x0 (0)

**Description**

The smoothed round trip time-out base minimum value for TCP round trip retransmission time in 1/2 second units. (Enter hexadecimal values.)

### **cfgNetTuningTcpSrttDflt (Read/Write)**

#### **Legal Values**

Integer from 6 to 384.

#### **Default**

0x6 (6)

#### **Description**

The smoothed round trip time-out base default value for TCP retransmission round trip time in 1/2 second units. (Enter hexadecimal values.)

### **cfgNetTuningTcpReXmtMin (Read/Write)**

#### **Legal Values**

Integer from 2 to 1024.

#### **Default**

0x2 (2)

#### **Description**

The minimum number of TCP retransmission time in 1/2 second units. (Enter hexadecimal values.)

### **cfgNetTuningTcpReXmtMax (Read/Write)**

#### **Legal Values**

Integer from 8 to 1024.

#### **Default**

0x80 (128)

#### **Description**

The maximum number of TCP retransmission time in 1/2 second units. (Enter hexadecimal values.)

### **cfgNetTuningTcpMsl (Read/Write)**

#### **Legal Values**

Integer from 60 to 240.

**Default**

0x3c (60)

**Description**

The maximum TCP segment lifetime in 1/2 second units. (Enter hexadecimal values.)

**Tuning the RAC for Satellite Connectivity**

The racadm utility may be used to modify the network tuning properties of the RAC. It is also possible to use the racadm.cfg file to modify these properties (similar to the manner in which .ini files are used).

```
racadm config -g cfgNetTuning -o cfgNetTuningNicMtu <value>
```

```
racadm config -g cfgNetTuning -o cfgNetTuningIpTtl <value>
```

```
racadm config -g cfgNetTuning -o cfgNetTuningIpReassTtl <value>
```

```
racadm config -g cfgNetTuning -o cfgNetTuningTcpSrttBase <value>
```

```
racadm config -g cfgNetTuning -o cfgNetTuningTcpSrttDflt <value>
```

```
racadm config -g cfgNetTuning -o cfgNetTuningTcpReXmtMin <value>
```

```
racadm config -g cfgNetTuning -o cfgNetTuningTcpReXmtMax <value>
```

```
racadm config -g cfgNetTuning -o cfgNetTuningTcpMsl <value>
```

where <value> is obtained from Table 2-1.

**Table 2-1. RAC Properties for Satellite Connectivity**

Property	Normal Value	Satellite Value
cfgNetTuningNicMtu	0x5dc	0x1f4
cfgNetTuningIpTtl	0x40	0x80
cfgNetTuningIpReassTtl	0x3c	0x78
cfgNetTuningTcpSrttBase	0	0x100
cfgNetTuningTcpSrttDflt	0x6	0x180
cfgNetTuningTcpReXmtMin	0	0
cfgNetTuningTcpReXmtMax	0x80	0x400
cfgNetTuningTcpMsl	0x3c	0xf0



**NOTICE:** Although these parameters can be configured, it is recommended that you not use any other settings than those detailed here. Other settings may adversely effect the RAC's ability to communicate with other network nodes.

Following the modification of the network tuning, the RAC must be reset for the new tuning values to take effect. After the RAC has been reset, it should be available for use in a normal or satellite network configuration.

## **cfgOobSnmpp**

The group contains parameters to configure the SNMP agent and trap capabilities of the RAC. One instance of the group is allowed. All objects require a RAC reset before they become active. The following subsections describe the objects in this group.

### **cfgOobSnmppTrapsEnable (Read/Write)**

#### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE)

#### **Default**

1

#### **Description**

0=Disable, 1=Enable transmission of SNMP traps.

### **cfgOobSnmppAgentEnable (Read/Write)**

#### **Legal Values**

Boolean either 1 or 0 (TRUE or FALSE).

#### **Default**

0

#### **Description**

0=Disable, 1=Enable the RAC SNMP agent.

## **cfgRacTuning**

The group contains various tuning configuration parameters.

One instance of the group is allowed. The following subsections describe the objects in this group.

**cfgRacTuneMnNwlpAddrBase (Read/Write)****Legal Values**

A string of "." separated numeric fields containing the IP address.

**Default**

192.168.234.235

**Description**

A base of two IP addresses used for the PPP connection to the RAC. This object requires a RAC reset before it becomes active.

**cfgRacTuneHostVncPort (Read/Write)****Legal Values**

5951

**Default**

5981

**Description**

The fixed port number for the client connection. This object requires a RAC reset before it becomes active. (Enter hexadecimal values.)

**cfgRacTuneGenericFtpUser (Read/Write)****Legal Values**

A string of up to 15 ASCII characters.

**Default**

racadm

**Description**

The user name used in FTP authentication.

**cfgRacTuneGenericFtpPasswd (Read/Write)****Legal Values**

A string of up to 15 ASCII characters.

**Default**

racadm

**Description**

The user password.

**cfgRacTuneRemoteCrsPort (Read/Write)****Legal Values**

Integer from 5900 to any positive valid port number.

**Default**

0x45B86000 or 5900

**Description**

The port connection from the browser to the firmware console redirection server. This object requires a RAC reset before it becomes active. (Enter hexadecimal values.)

**cfgRacTuneFwUpdateResetDelay (Read/Write)****Legal Values**

Integer from 0 to 600.

**Default**

0x46 or 70

**Description**

The maximum number in seconds to wait between when the firmware update file is loaded, and the update sequence is started. (Enter hexadecimal values.)

**cfgRacTuneDialOutIdleTime (Read/Write)****Legal Values**

Integer from 0 to any positive value.

**Default**

0x28 or 40

**Description**

0=Infinite, otherwise when line is idle for this many seconds, the line is disconnected. This object requires a RAC reset before it becomes active. (Enter hexadecimal values.)

**cfgRacTuneDialOutMaxConnectTime (Read/Write)****Legal Values**

Integer from 1 to 128.

**Default**

0x0

**Description**

0=Infinite, otherwise maximum time in seconds that the link will stay connected before the connection is terminated. This object requires a RAC reset before it becomes active. (Enter hexadecimal values.)

**cfgRacTuneD3debugEnable (Read/Write)****Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

**Default**

1

**Description**

0=disable, 1=enable the RAC debug command. This object requires a RAC reset before it becomes active.

**cfgRacTuneRemoteRacadmEnable (Read/Write)****Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

**Default**

0

**Description**

0=Disable, 1=Enable

This object requires a RAC reset before it becomes active.

## **cfgRacTuneMnConsoleDisableAllowed**

### **Legal Values**

Boolean, either 1 or 0 (TRUE or FALSE).

### **Default**

FALSE

### **Description**

This command is applicable only to Microsoft® Windows® 2000 and Windows Server 2003 operating systems. When **TRUE**, a user performing remote console redirection through the RAC Web-based remote access interface can disable the managed system's keyboard and mouse. When **FALSE**, the managed system's keyboard and mouse cannot be disabled.

## **cfgRacTuneHostCom2BaudRate (value)**

### **Default**

57600

### **Legal Values**

115200, 57600, 19200, and 9600



**NOTE:** For best results when redirecting BIOS setup screens, Dell recommends setting this baud rate to 57600.

## **ifcRacManagedNodeOs**

This group contains parameters to configure the RAC with managed system and operating system naming information.

One instance of the group is allowed. The following subsections describe the objects in this group.

### **ifcRacMnOsHostname (Read/Write)**

#### **Legal Values**

A string of up to 255 ASCII characters.

#### **Default**

""

#### **Description**

The host name of the managed system.

**ifcRacMnOsOsName (Read/Write)****Legal Values**

A string of any ASCII characters.

**Default**

""

**Description**

The operating system name of the managed system.

**ifcRacMnOsOsType (Read Only)****Legal Values**

Not user-writable.

**Default**

0

**Description**

See the getsysinfo command.

**cfgRacSecurity**

This group contains parameters to configure the RAC SSL (Secure Sockets Layer) security features.

**cfgRacSecCapabilities (Read Only)****Legal Values**

0x1ff

**Default**

0x1ff

**Description**

The current RAC security capability setting.

### **cfgRacSecCsrKeySize (Read Only)**

#### **Legal Values**

512 or 1024

#### **Default**

1024

#### **Description**

The Web SSL handshake RSA key size.

### **cfgRacSecCsrCommonName (Read/Write)**

#### **Legal Values**

A string of any ASCII characters

#### **Default**

""

#### **Description**

The CSR (certificate signing request) common name.

### **cfgRacSecCsrOrganizationName (Read/Write)**

#### **Legal Values**

A string of any ASCII characters.

#### **Default**

""

#### **Description**

The CSR organization name.

### **cfgRacSecCsrOrganizationUnit (Read/Write)**

#### **Legal Values**

A string of any ASCII characters.

#### **Default**

""

**Description**

The CSR organization unit.

**cfgRacSecCsrLocalityName (Read/Write)****Legal Values**

A string of any ASCII characters.

**Default**

""

**Description**

The CSR locality name.

**cfgRacSecCsrStateName (Read/Write)****Legal Values**

A string of any ASCII characters.

**Default**

""

**Description**

The CSR state.

**cfgRacSecCsrCountryCode (Read/Write)****Legal Values**

The two-letter country code.

**Default**

""

**Description**

The CSR country code.

**cfgRacSecCsrEmailAddr (Read/Write)****Legal Values**

A string of any ASCII characters.

**Default**

""

**Description**

The CSR e-mail address.

**cfgRacSecVncInEncryptEnable (Read/Write)****Legal Values**

Boolean, either TRUE or FALSE.

**Default**

"TRUE"

**Description**

Enables VNC (virtual network computing) console redirection encryption.

FALSE=Disable, TRUE=Enable

This object requires a RAC reset before it becomes active.

**cfgRacSecAuthLocalRacEnable (Read Only)****Legal Values**

0x01

**Default**

0x01

**Description**

Enables RAC local base authentication.

**cfgRacSecAuthLocalOsEnable (Read/Write)****Legal Values**

Boolean, either TRUE or FALSE.

**Default**

"TRUE"

**Description**

Enables RAC operating system base authentication.

**FALSE**=Disable, **TRUE**=Enable

This object requires a RAC reset before it becomes active.

# Abbreviations and Acronyms

## **AGP**

Abbreviation for accelerated graphics port, which is a bus specification that allows graphics cards faster access to main system memory.

## **ANSI**

Abbreviation for American National Standards Institute.

## **API**

Abbreviation for application program interface, which is a set of routines, protocols, and tools for building software applications that act as an interface between the operating system and application programs.

## **ARP**

Acronym for Address Resolution Protocol, which is a method for finding a host's Ethernet address from its Internet address.

## **ASCII**

Acronym for American Standard Code for Information Interchange, which is a code representation used for displaying or printing letters, numbers, and other characters.

## **BIOS**

Acronym for basic input/output system, which is the part of system software that provides the lowest-level interface to peripheral devices and which controls the first stage of the system boot process, including installation of the operating system into memory.

## **BMC**

Abbreviation for baseboard management controller, which is the controller interface between the RAC and the managed system's ESM.

## **bus**

A set of conductors connecting the various functional units in a computer. Busses are named by the type of data they carry, such as data bus, address bus, or PCI bus.

## **CD**

Abbreviation for compact disc.

## **CHAP**

Acronym for Challenge-Handshake Authentication Protocol, which is an authentication method used by PPP servers to validate the identity of the originator of the connection.

## **CLI**

Abbreviation for command-line interface.

## **console redirection**

Console redirection is a function that directs a managed system's display screen, mouse functions, and keyboard functions to the corresponding devices on a management station. You may then use the management station's system console to control the managed system.

## **CSR**

Abbreviation for customer service representative.

**DHCP**

Abbreviation for Dynamic Host Configuration Protocol, which is a protocol that provides a means to dynamically allocate IP addresses to computers on a local area network.

**DRAC III**

Abbreviation for Dell Remote Access Card III.

**DRAC III/XT**

Abbreviation for Dell Remote Access Card III/XT.

**DSU**

Abbreviation for disk storage unit.

**ERA**

Abbreviation for Embedded Remote Access.

**ERA/O**

Abbreviation for Embedded Remote Access Option.

**ESM**

Abbreviation for embedded systems management, which is a set of instruction coding in system software and firmware that notifies a user about potential hardware problems on a Dell system.

**GMT**

Abbreviation for Greenwich Mean Time, which is the standard time common to every place in the world. GMT nominally reflects the mean solar time along the prime meridian (0 longitude) that runs through the Greenwich Observatory outside of London, UK.

**GUI**

Abbreviation for graphical user interface, which refers to a computer display interface that uses elements such as windows, dialog boxes, and buttons as opposed to a command prompt interface, in which all user interaction is displayed and typed in text.

**hardware log**

Records events generated by the RAC and ESM.

**ICMB**

Abbreviation for Intelligent Chassis Management Bus.

**ID**

Abbreviation for identifier, commonly used when referring to a user identifier (user ID) or object identifier (object ID).

**IP**

Abbreviation for Internet Protocol, which is the network layer for TCP/IP. IP provides packet routing, fragmentation, and reassembly.

**IPMB**

Abbreviation for intelligent platform management bus, which is a bus used in systems management technology.

**IPMI**

Abbreviation for Intelligent Platform Management Interface, which is a part of systems management technology.

**Kbps**

Abbreviation for kilobits per second, which is a data transfer rate.

**LAN**

Abbreviation for local area network.

**LED**

Abbreviation for light-emitting diode.

**MAC**

Acronym for media access control, which is a network sublayer between a network node and the network physical layer.

**MAC address**

Acronym for media access control address, which is a unique address embedded in the physical components of a network adapter.

**Mbps**

Abbreviation for megabits per second, which is a data transfer rate.

**MIB**

Abbreviation for management information base.

**PCI**

Abbreviation for Peripheral Component Interconnect, which is a standard interface and bus technology for connecting peripherals to a system and for communicating with those peripherals.

**PCMCIA**

Abbreviation for Personal Computer Memory Card International Association, which is a technology standard for connecting PCMCIA devices to systems. PCMCIA devices are about the size of a credit card and are mainly used in notebook computers, although the DRAC III supports an optional PCMCIA modem.

**POST**

Acronym for power-on self-test, which is a sequence of diagnostic tests that are run automatically by a system when it is powered on.

**PPP**

Abbreviation for Point-to-Point Protocol, which is the Internet standard protocol for transmitting network layer datagrams (such as IP packets) over serial point-to-point links.

**RAC**

Abbreviation for remote access controller.

**RAM**

Acronym for random-access memory. RAM is general-purpose readable and writable memory on systems and the RAC.

**RAM disk**

A memory-resident program which emulates a hard drive. The RAC maintains a RAM disk in its memory.

**RAS**

Acronym for remote access service, which is a service provided by some Microsoft® Windows® operating systems that allows most of the services that would be available on a network to be accessed over a modem. The service includes support for dial-up and logon tasks and presents the same network interface as the normal network drivers used for the network adapter.

**RFB**

Abbreviation for remote floppy boot.

**ROM**

Acronym for read-only memory, which is memory from which data may be read, but to which data cannot be written.

**RPM**

Abbreviation for Red Hat Package Manager, which is a package-management system for Red Hat Linux that helps installation of software packages. It is similar to an installation program.

**RSA**

Abbreviation for Rivest, Shamir, and Adleman, the inventors of a public key encryption technology developed by RSA Data Security, Inc.

**SNMP**

Abbreviation for Simple Network Management Protocol, which is a protocol designed to manage nodes on an IP network. RACs are SNMP-managed devices (nodes).

**SNMP trap**

A notification (event) generated by the RAC or ESM that contains information about state changes on the managed system or about potential hardware problems.

**SSL**

Acronym for Secure Sockets Layer, a protocol designed by Netscape to provide encrypted,

authenticated communication across the Internet. SSL is used primarily for sending communications between Web browsers and Web servers. Web addresses using an SSL connection begin with "https."

**TAP**

Abbreviation for Telelocator Alphanumeric Protocol, which is a protocol used for submitting requests to a pager service.

**TCP/IP**

Abbreviation for Transmission Control Protocol/Internet Protocol, which represents the set of standard Ethernet protocols that includes the network layer and transport layer protocols.

**TFTP**

Abbreviation for Trivial File Transfer Protocol, which is a simple file transfer protocol used for downloading boot code to diskless devices or systems.

**UPS**

Abbreviation for uninterruptible power supply.

**UTC**

Abbreviation for Universal Coordinated Time. *See* GMT.

**VNC**

Abbreviation for virtual network computing.

**VT-100**

Abbreviation for Video Terminal 100, which is used by the most common terminal emulation programs.

**WAN**

Abbreviation for wide area network.