uCPE Networking DIAG OS Guide

April 2020

DELL EMC

Notes, cautions, and warnings

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

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

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

2020 - 04

Rev. A04

Contents

1

About this guide

Notices

CAUTION: To avoid electrostatic discharge (ESD) damage, wear grounding wrist straps when handling this equipment.

NOTE: Only trained and qualified personnel can install this equipment. Read this guide before you install and power up this equipment. This equipment contains two power cords. Disconnect both power cords before servicing.

NOTE: This equipment contains optical transceivers, which comply with the limits of Class 1 laser radiation.

Figure 1. Class 1 laser product tag

NOTE: When no cable is connected, visible and invisible laser radiation may be emitted from the aperture of the optical transceiver ports. Avoid exposure to laser radiation and do not stare into open apertures.

Related documents

For more information about your Open Networking (-ON) platform, see the following documents:

- *Dell EMC Getting Started Guide or Dell EMC Setup Guide*
- *Dell EMC Installation Guide*
- *Dell EMC Release Notes*

2

BIOS setup and configuration

This section describes how to access the BIOS setup and configuration screen on your system.

Access the BIOS setup and configuration screen from the command prompt. Ensure that your TFTP server is reachable over your network.

NOTE: The following output examples are for reference only; your output may vary.

NOTE: The management port IP, FTP server IP address, MAC address, and user-id shown are for illustration purpose only. Use your system's applicable values.

Topics:

Power on VEP4600

Plug in a power cord to the back of VEP4600 platform. The platform starts to power up immediately.

Create a serial console connection

To establish a console connection use a universal serial bus (USB)-to-RS-232 connection from a USB port to a VEP4600 console port.

NOTE: Use a 115200 baud rate.

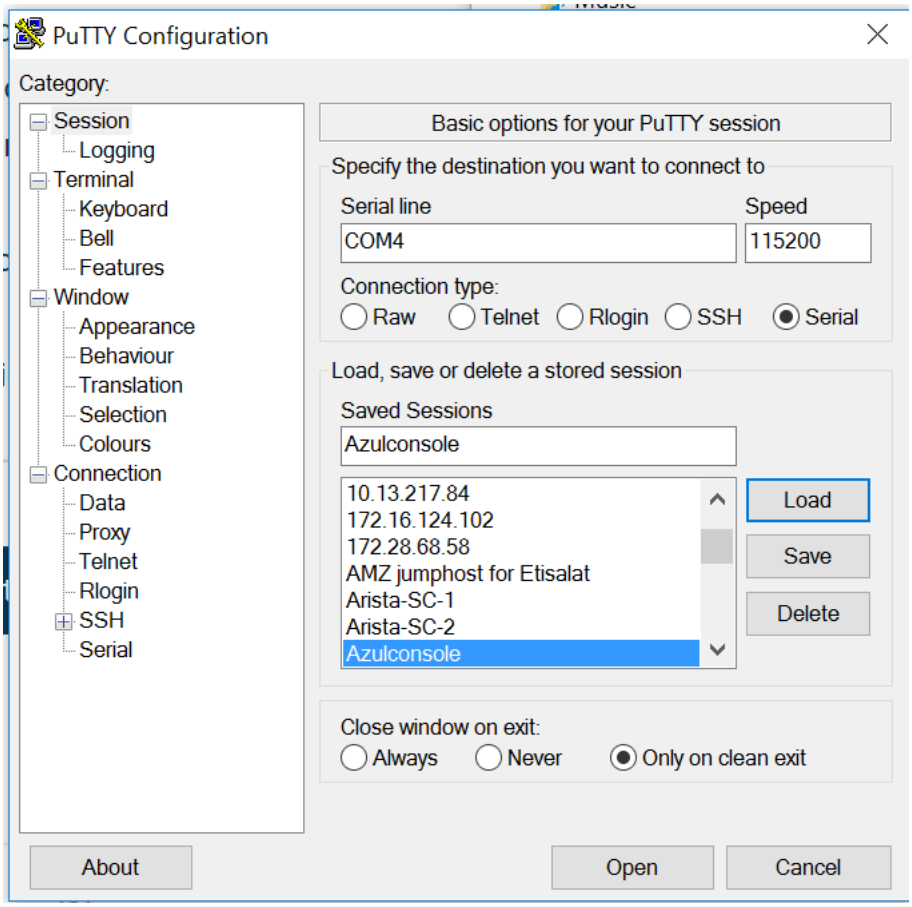

Figure 2. puTTY 115200 baud rate setup

BIOS access process

1. Press the **delete** button after the `POST Lower DRAM Memory test` appears on the screen.

Continue pressing the **delete** button to progress to the `BIOS setup and configuration` screen.

NOTE: If the `BIOS setup and configuration` screen window passes, power off and power on the platform again to restart the boot up process.

```
CPLD Reset Source=0x44


POST Configuration
  CPU Signature 50654
  CPU FamilyID=6, Model=55, SteppingId=4, Processor=0
  Microcode Revision 2000043
  Platform ID: 0x10000000000000
  PKG_CST_CFG_CTL: 0x3
  Misc EN: 0x4000840088
  Gen PM Con1: 0x0
  Therm Status: 0x8000000
  POST Control=0xEA000303, Status=0xE6008500

BIOS initializations...

POST:
  RTC Battery OK at last cold boot
  RTC date 5/4/2018 3:02:03

POST SPD test ..............................  PASS

POST Lower DRAM Memory test
...
```

Figure 3. Initial boot up screen

```
Version 2.20.1271. Copyright (C) 2018 American Megatrends, Inc.
BIOS Date: 04/11/2018 02:44:05 Ver: 0ACJF020
Press <DEL> or <F2> to enter setup.
```

Figure 4. Boot up screen

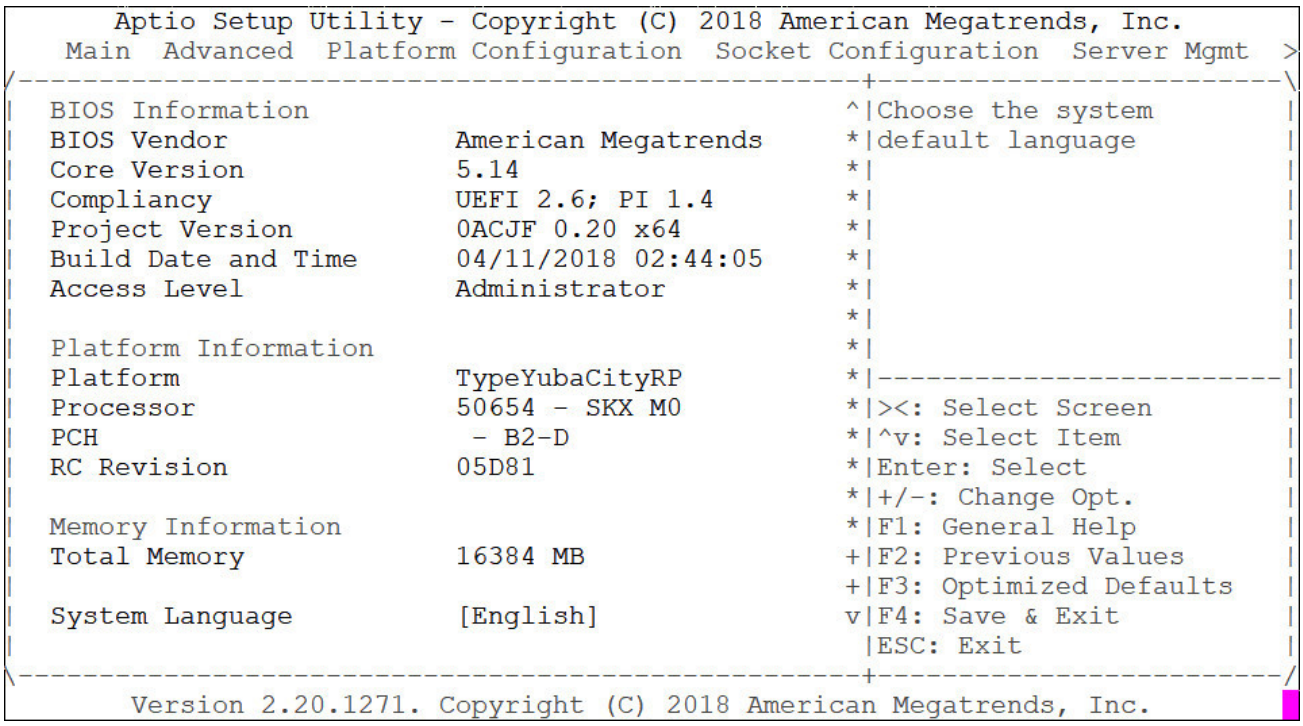

Figure 5. BIOS setup and configuration screen

3

Dell EMC DIAG OS

The following describes the Dell EMC diagnostics operating system (DIAG OS).

Topics:

View DIAG versions

To display the DIAG version installed in the DIAG OS, use the `dpkg -l | grep dn-diags` command at the `root@dell-diag-os:~` prompt.

```
root@dell-diag-os:/# dpkg -l | grep dn-diags
ii  dn-diags-<platform>-on.deb  1.10 amd64   Dell Diagnostics
root@dell-diag-os:/#
```

4

Restore to Manufacture DIAG OS

Manufacture DIAG OS recovery for the VEP4600 platform.

Burn DIAG OS ISO image to a bootable USB

1. Mount the USB to a Linux computer or VEP4600 with DIAG OS.
2. Log in to the Linux OS.
3. Download the DIAG OS ISO image to the Linux computer using TCP, SCP, or a similar protocol.
4. Use the following DD (data duplicator) CLI (command line interface) Linux command to copy the DIAG OS to the USB.

```
dd if=onie-recovery-x86_64-dellemc_vep4600_d21xyt-r0.48.iso of=/dev/sdb bs=4M
```

NOTE: Use `/dev/sdb`, not the `sdb#` number even if the disk shows `sdb#` as one of the USB sticks plugged in.

```
Device     Boot Start   End Sectors  Size Id Type
/dev/sdb1  *     7516  7899     384  192K ef EFI (FAT-12/16/32)
```

Configure BIOS to install DIAG OS from USB

1. Boot into `BIOS` setting, goto `Advanced`, set `CSM` to `UEFI only`

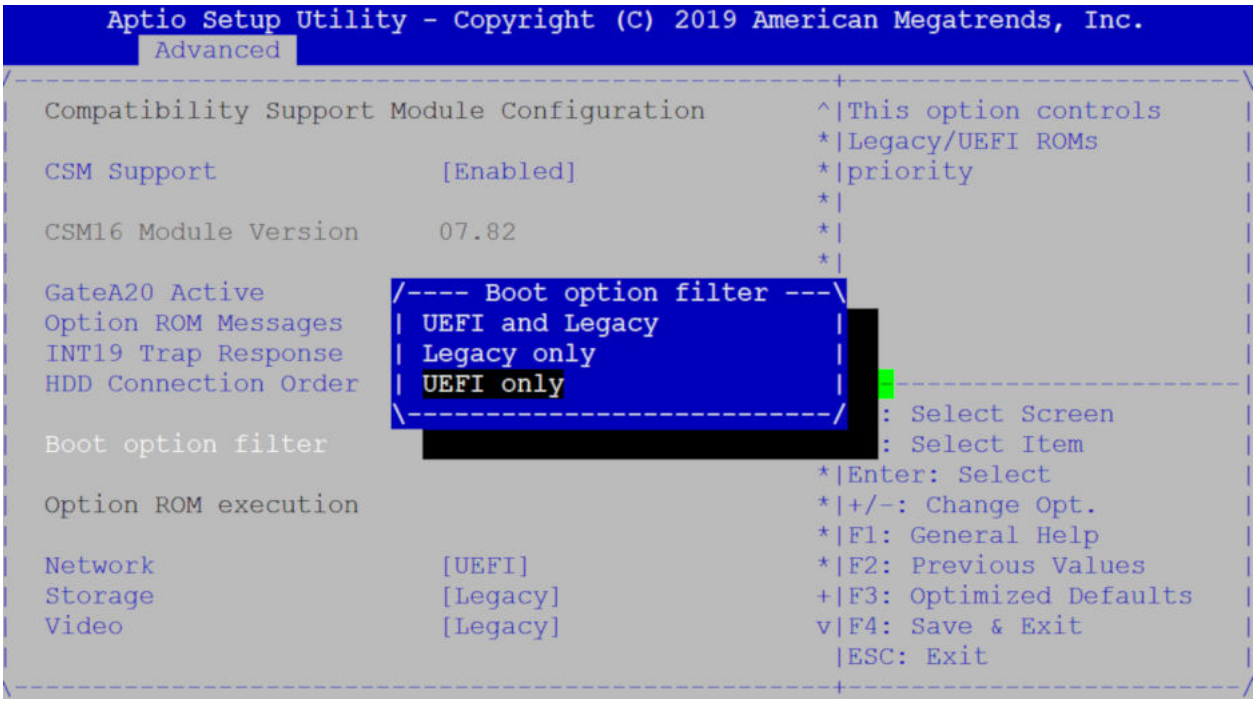

Figure 6. Boot BIOS setting

2. Select the **Boot** menu tab.

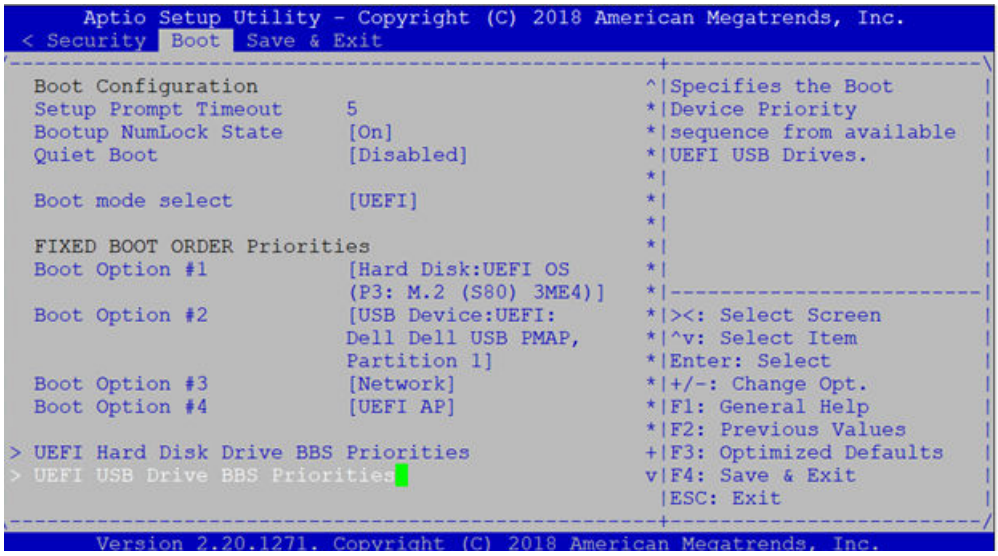

Figure 7. Boot menu tab

3. Select **UEFI:** then **USB Device:** to boot the DIAG OS from a USB drive.

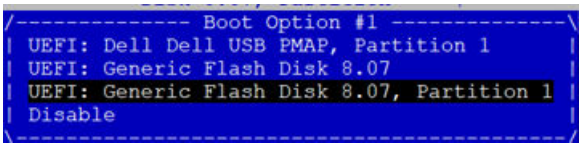

Figure 8. DIAG OS USB to boot UEFI

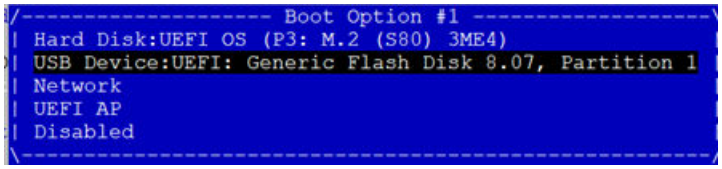

Figure 9. DIAG OS USB to boot USB device

4. Verify that **Boot Option #1** lists the DIAG OS USB as the boot option.

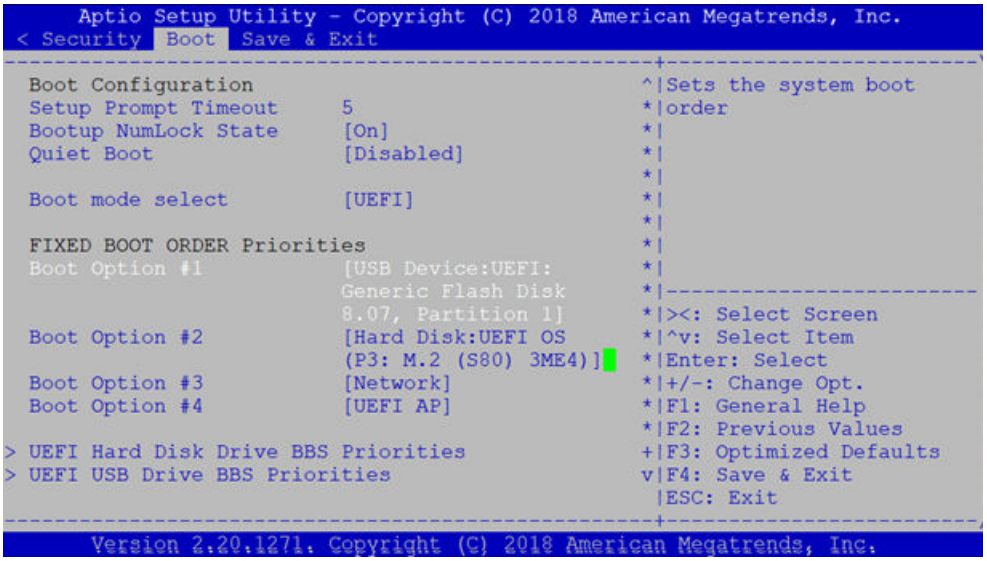

Figure 10. Boot Option #1

5. Press **F4** to save and exit the utility and to start the installation.

DIAG OS installation failure and resolution

ESXi may create a different disk partition that is not compatible with the DIAG OS. This causes the DIAG OS installation to fail and display this error message:

```
ONIE: Rescue Mode ...
Platform  : x86_64-dellemc_vep4600_d21xyt-r0
Version   : x.xx.x.xx-x
Build Date: 2018-04-24T03:20-0700
▒[   13.793445] ata4.00: failed to set xfermode (err_mask=0x40)
Info: Mounting kernel filesystems... done.
Info: Using eth0 MAC address: d8:9e:f3:bc:6a:a0
Info: eth0:  Checking link... up.
Info: Trying DHCPv4 on interface: eth0
Warning: Unable to configure interface using DHCPv4: eth0
ONIE: Using link-local IPv4 addr: eth0: xxx.xxx.x.xxx/xx
+ cat /DiagOS_version.cfg
+ version_packed=x.xx.x.xx-x
+ ls
+ grep x.xx.x.xx-x
+ image_packed=diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-x-2018-04-24.bin
+ [ -z diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-x-2018-04-24.bin ]
+ echo starting to install vep4600 DiagOS
starting to install vep4600 DiagOS
+ onie-nos-install /diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-x-2018-04-24.bin
discover: Rescue mode detected. No discover stopped.
ONIE: Executing installer: /diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-
x-2018-04-24.bin
Ignoring Verifying image checksum ... OK.
cur_dir / archive_path /var/tmp/installer tmp_dir /tmp/tmp.XeWxoj
Preparing image archive ...sed -e '1,/^exit_marker$/d' /var/tmp/installer | tar xf - OK.
Diag-OS Installer: platform: x86_64-dellemc_vep4600_d21xyt-r0

EDA-DIAG Partiton not found.
Diag OS Installer Mode : INSTALL

Deleting partition at /dev/sdc1...
The operation has completed successfully.
Deleting partition at /dev/sdc2...
The operation has completed successfully.
Deleting partition at /dev/sdc3...
The operation has completed successfully.
Deleting partition at /dev/sdc4...
Partition number 4 out of range!
Error 0 deleting partition!
Error encountered; not saving changes.
Error: Unable to delete partition 4 on /dev/sdc
Removing /tmp/tmp.XeWxoj
Failure: Unable to install image: /diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-
x-2018-04-24.bin
+ echo This should be not reachable unless something wrong is there!!!!!
This should be not reachable unless something wrong is there!!!!!
Starting: dropbear ssh daemon... done.
Starting: telnetd... done.
discover: Rescue mode detected.  Installer disabled.

Please press Enter to activate this console.
To check the install status inspect /var/log/onie.log.
Try this:  tail -f /var/log/onie.log
```

To resolve this issue, delete the partition completely and restart the DIAG OS installation.

1. Press **Enter** from the error message to get to ONIE Recovery mode.

   NOTE: WARNING: Deleting the partition causes all data and the OS to be lost.

2. Type the following then click **Enter**.

```
gdisk /dev/sdc
```

```
** Rescue Mode Enabled **
ONIE-RECOVERY:/ #

gdisk /dev/sdc
GPT fdisk (gdisk) version 0.8.8
Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
Found valid GPT with protective MBR; using GPT.
```

3. Type `o` to delete the partition.

```
Command (? for help):

Command (? for help): o
This option deletes all partitions and creates a new protective MBR.
Proceed? (Y/N): y

Type w to write the new partition into the disk

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/sdc.
The operation has completed successfully.
ONIE-RECOVERY:/ #
```

4. Type `reboot` at the command prompt and restart the DIAG OS installation. A successful installation displays the following:

```
ONIE: Rescue Mode ...
Platform  : x86_64-dellemc_vep4600_d21xyt-r0
Version   : x.xx.x.xx-x
Build Date: 2018-04-24T03:20-0700
▒[   12.771519] ata4.00: failed to set xfermode (err_mask=0x40)
Info: Mounting kernel filesystems... done.
Info: Using eth0 MAC address: d8:9e:f3:bc:6a:a0
Info: eth0:  Checking link... up.
Info: Trying DHCPv4 on interface: eth0
Warning: Unable to configure interface using DHCPv4: eth0
ONIE: Using link-local IPv4 addr: eth0: 169.254.195.48/16
+ cat /DiagOS_version.cfg
+ version_packed=x.xx.x.xx-x
+ ls
+ grep x.xx.x.xx-x
+ image_packed=diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-x-2018-04-24.bin
+ [ -z diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-x-2018-04-24.bin ]
+ echo starting to install vep4600 DiagOS
starting to install vep4600 DiagOS
+ onie-nos-install /diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-
x-2018-04-24.bin
discover: Rescue mode detected. No discover stopped.
ONIE: Executing installer: /diag-installer-x86_64-dellemc_vep4600_d21xyt-r0-x.xx.x.xx-
x-2018-04-24.bin
Ignoring Verifying image checksum ... OK.
cur_dir / archive_path /var/tmp/installer tmp_dir /tmp/tmp.yb6fIB
Preparing image archive ...sed -e '1,/^exit_marker$/d' /var/tmp/installer | tar xf - OK.
Diag-OS Installer: platform: x86_64-dellemc_vep4600_d21xyt-r0

EDA-DIAG Partiton not found.
Diag OS Installer Mode : INSTALL

partprobe in remove all partitions
GPT data structures destroyed! You may now partition the disk using fdisk or
```

```
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
The operation has completed successfully.
The operation has completed successfully.
mkfs.fat 3.0.26 (2014-03-07)
create_grub_boot_partition finished !
Creating new diag-os partition /dev/sdc2 ...
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot.
The operation has completed successfully.

EDA-DIAG dev is /dev/sdc2
mke2fs 1.42.13 (17-May-2015)
Discarding device blocks: done
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: c7971d6a-acb1-46be-84a2-a8d2d758139b
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

Created filesystem on /dev/sdc2 with label EDA-DIAG

Mounted /dev/sdc2 on  /tmp/tmp.iK7Bg3

Preparing /dev/sdc2 EDA-DIAG for rootfs install
untaring into /tmp/tmp.iK7Bg3

rootfs copy done
Success: Support tarball created: /tmp/tmp.iK7Bg3/onie-support.tar.bz2
Updating Grub Cfg /dev/sdc2 EDA-DIAG
```

Configure BIOS and boot into DIAG OS

After the DIAG OS installation completes, configure the BIOS then boot into the DIAG OS.

1. Boot into the BIOS setting.
2. Configure **Boot Option #1** from the **Boot Configuration** screen.

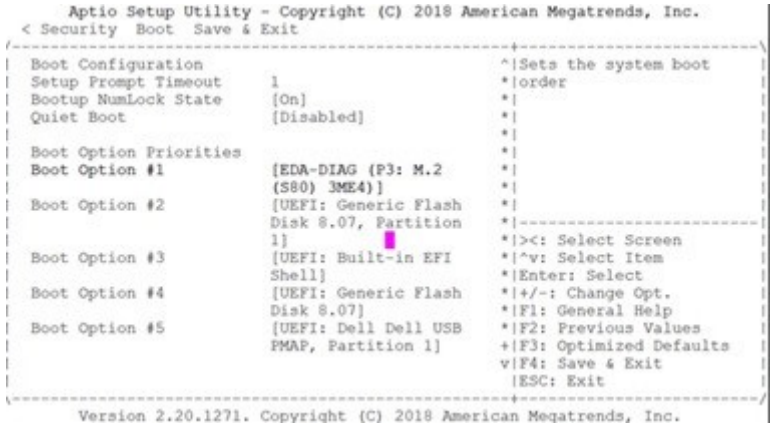

Figure 11. Boot configuration screen

3. Press the **F4** key to save the changes and exit the utility.
4. Confirm saving the configuration using the left and right arrow keys, and exit from the utility. Select **Yes** and press **Enter**.

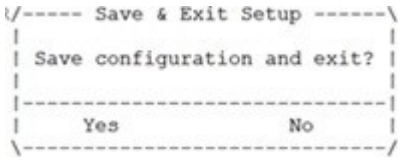

Figure 12. Save & exit

After you save the changes the log in command prompt displays.

Figure 13. Log in command prompt

5. Type to log in.

```
root/calvin
```

DIAG OS Verification

NOTE: The system shows the current version.

After DIAG OS installation, to verify the DIAG OS version, boot into boot into the DIAG OS by the following commands.

1. Log in into the DIAG OS using root as the `username` and `calvin`as the password.
2. Enter the `sh_ver` command.

```
root@dellemc-diag-os:~#sh_ver
Diag OS version VEP4600_DIAG_OS_x.xx.x.xx-x
Build date/time Tue Apr 24 00:15:20 PDT 2018
Build server    netLogin-eqx-03
Build by        cwang3
Kernel Info:
Linux 4.9.30 #1 SMP PREEMPT Tue Apr 24 00:12:19 PDT 2018 x86_64 GNU/Linux
Debian GNU/Linux 8 \n \l
root@dellemc-diag-os:~#
```

5

Dell EMC DIAG OS tools

This section describes how to use the Dell EMC diagnostics operating system (DIAG OS). The DIAG OS provides a suite of tools to help diagnose issues seen on the system, or to run a health check to ensure that the hardware is operating properly.

Diagnostic tools

The DIAG OS uses standard Linux drivers and contains the following tools you can use to evaluate the health of your system. The tools are packaged for both the DIAG OS, which is a simple OS of the same kernel version, and small `rootfs` to support the tools and drivers.

Topics:

edatool

The `edatool` is included in the diagnostic tools. Use the tool to test the basic functionality of the system.

The `edatool` executes a script of simple commands, similar to commands in the CLI. Usually, the diagnostics tools run these types of tests. The success or failure of these tools is reported, and at the end of the `edatool` run, reports the PASSED or FAILED results in a standard format the test scripts can easily parse.

Tests

The `edatool` does not have a test command, but instead runs all the tests that are scripted.

CLI options

```
DellEmc Diag - Extended Diagnostics Application
version 1.4, x.xx.x.x-x
build, 2017/05/23,
```

```
Syntax: edatool <option>
 Show the Help-text:=
        edatool --h                                                    (or)
        edatool -h
 Lists tests in config files:=
        edatool --list                                                 (or)
        edatool -l
 Config file to use for tests:=
        edatool --config=<config_file>                                 (or)
        edatool -f <config_file>
 Config file to use for extended tests:=
        edatool --extended-config=<config_file>                        (or)
        edatool -X <config_file>
 Display test list or test result or modify test item status:=
        edatool --testlist=show/result/<on/off,<test_id>,<test_id>...>(or)
        edatool -L show/result/<on/off,<test_id>,<test_id>...>
 Run all or selected test item in test list:=
        edatool --testrun=all/<test_id>                                (or)
        edatool --R all/<test_id>
 Execute repeatedly command by count:=
        edatool --iteration=max/<count> [option1] [option2]...         (or)
        edatool -I max/<count> [option1] [option2]...

Usage:=
  -h, --h                 Show the help text
  -l, --list              List the understood TLV codes and names
  -I, --iteration=        Iteration command execution
  -L, --testlist=         Test list status
  -R, --testrun=          Run test item
  -f, --config=           To specify the location of the config file e.g. /etc/dn/diag/
<file_name>
  -X, --extended-config=  Config file to use for extended tests
```

Output

```
root@dell-diag-os:~# edatool
*****************************
*  Diagnostics Application  *
*****************************
Dell-EMC Diag edatool version 1.4, package x.xx.x.x 2016/11/21
Dell-EMC Diag cputool - version 1.1 package x.xx.x.x 2016/11/21
Dell-EMC Diag fantool - version 1.5 package x.xx.x.x 2016/11/21
Dell-EMC Diag gpiotool - version 1.4 package x.xx.x.x 2016/11/21
Dell-EMC Diag i2ctool - version 1.5 package x.xx.x.x 2016/11/21
Dell-EMC Diag ledtool - version 1.0 package x.xx.x.x 2016/11/21
Dell-EMC Diag lpctool - version 1.0 package x.xx.x.x 2016/11/21
Dell-EMC Diag memtool - version 1.5 package x.xx.x.x 2016/11/21
Dell-EMC Diag nputool - version 1.0 sdk-6.5.5 package x.xx.x.x 2016/11/21
Dell-EMC Diag nvramtool - version 1.5 package x.xx.x.x 2016/11/21
Dell-EMC Diag opticstool - version 1.0 package x.xx.x.x 2016/11/21
Dell-EMC Diag pcitool - version 1.5 package x.xx.x.x 2016/11/21
Dell-EMC Diag pltool - version 1.5 package x.xx.x.x 2016/11/21
Dell-EMC Diag psutool - version 1.4 package x.xx.x.x 2016/11/21
Dell-EMC Diag rtctool - version 1.1 package x.xx.x.x 2016/11/21
Dell-EMC Diag smbiostool - version 1.2 package x.xx.x.x 2016/11/21
Dell-EMC Diag storagetool - version 1.1 package x.xx.x.x 2016/11/21
Dell-EMC Diag temptool - version 1.4 package x.xx.x.x 2016/11/21

Testing PCI devices:
+ Checking PCI 00:00.0, ID=1f0c8086 ..................... Passed
+ Checking PCI 00:01.0, ID=1f108086 ..................... Passed
+ Checking PCI 00:02.0, ID=1f118086 ..................... Passed
+ Checking PCI 00:03.0, ID=1f128086 ..................... Passed
+ Checking PCI 00:0e.0, ID=1f148086 ..................... Passed
+ Checking PCI 00:0f.0, ID=1f168086 ..................... Passed
+ Checking PCI 00:13.0, ID=1f158086 ..................... Passed
+ Checking PCI 00:14.0, ID=1f418086 ..................... Passed
+ Checking PCI 00:14.1, ID=1f418086 ..................... Passed
+ Checking PCI 00:14.2, ID=1f418086 ..................... Passed
+ Checking PCI 00:16.0, ID=1f2c8086 ..................... Passed
```

```
+ Checking PCI 00:17.0, ID=1f228086 ..................... Passed
+ Checking PCI 00:18.0, ID=1f328086 ..................... Passed
+ Checking PCI 00:1f.0, ID=1f388086 ..................... Passed
+ Checking PCI 00:1f.3, ID=1f3c8086 ..................... Passed
+ Checking PCI 01:00.0, ID=837514e4 ..................... Passed
+ Checking PCI 01:00.1, ID=837514e4 ..................... Passed
PCI devices: Overall test results -------------------- >>> Passed
Testing I2C devices:

Checking I2C devices on bus 0:

+ Checking Clock GEN             0x69  ..... Passed

+ Checking SPD0                  0x50  ..... Passed

Checking I2C devices on bus 1:

+ Checking CPU Board I2C Mux     0x70  ..... Passed

+ Checking CPU Board EEPROM1     0x53  ..... Passed

+ Checking CPU Board EEPROM2     0x57  ..... Passed

+ Checking Switch Brd EEPROM     0x50  ..... Passed

+ Checking CPLD2                 0x3e  ..... Passed

+ Checking CPLD3                 0x3e  ..... Passed

+ Checking CPLD4                 0x3e  ..... Passed

+ Checking SFP+ 1                  0x50  ..... Passed

+ Checking SFP+ 2                  0x50  ..... Passed

+ Checking SFP+ 3                  0x50  ..... Passed

+ Checking SFP+ 4                  0x50  ..... Passed

+ Checking SFP+ 5                  0x50  ..... Passed

+ Checking SFP+ 6                  0x50  ..... Passed

+ Checking SFP+ 7                  0x50  ..... Passed

+ Checking SFP+ 8                  0x50  ..... Passed

+ Checking SFP+ 9                  0x50  ..... Passed

+ Checking SFP+ 10                  0x50  ..... Passed

+ Checking SFP+ 11                 0x50  ..... Passed

+ Checking SFP+ 12                 0x50  ..... Passed

+ Checking SFP+ 13                 0x50  ..... Passed

+ Checking SFP+ 14                 0x50  ..... Passed

+ Checking SFP+ 15                 0x50  ..... Passed

+ Checking SFP+ 16                 0x50  ..... Passed

+ Checking SFP+ 17                 0x50  ..... Passed

+ Checking SFP+ 18                 0x50  ..... Passed

+ Checking SFP+ 19                 0x50  ..... Passed

+ Checking SFP+ 20                 0x50  ..... Passed

+ Checking SFP+ 21                 0x50  ..... Passed
```

```
+ Checking SFP+ 22                0x50  ..... Passed
+ Checking SFP+ 23                0x50  ..... Passed
+ Checking SFP+ 24                0x50  ..... Passed
+ Checking SFP+ 25                0x50  ..... Passed
+ Checking SFP+ 26                0x50  ..... Passed
+ Checking SFP+ 27                0x50  ..... Passed
+ Checking SFP+ 28                0x50  ..... Passed
+ Checking SFP+ 29                0x50  ..... Passed
+ Checking SFP+ 30                0x50  ..... Passed
+ Checking SFP+ 31                0x50  ..... Passed
+ Checking SFP+ 32                0x50  ..... Passed
+ Checking SFP+ 33                0x50  ..... Passed
+ Checking SFP+ 34                0x50  ..... Passed
+ Checking SFP+ 35                0x50  .....
```

Verbose mode

Use the following steps to enable and set the verbose level.

1. Set the Verbose level with a value of 0 to 3 using bits 4 and 5 of the EDA control reg (0x55).

   For example, to set the verbose level to 2, set bit 5 to 1 (`5=1`) and bit 4 to 0 (`4=0`).

   ```
   root@dellemc-diag-os:~# nvramtool --write --reg=0x55 --val=0x25
   ```

   The value is written in hexadecimal. The `xx10x1xx` shows the bit positions of 2, 4&5, and bit 0 on the right.
2. Enable Verbose mode by setting bit 2 of the same reg to `1`.

   NOTE: If you disable Verbose mode, or bit 2 of reg 0x55 is set to `0`, the default verbosity level is `0/zero`.

EDA control reg (0x55):

- 5:4—EDA Verbose Level = 0/1/2/3 or verbosity level 0, 1, 2, or 3.
- 3—EDA Extended Tests
- 2—EDA Verbose Mode = 0/1 (0=disabled; 1=enabled)
- 1—EDA Stop on Error
- 0—EDA Enable

NOTE: If you do not need the Verbose mode settings to persist through reboots, you can use the environment variable method to enable Verbose Mode.

```
export VERB_LEVEL=<setting 0,1,2 or 3>
```

To clear the environment variable, use the `unset VERB_LEVEL` command.

cputool

The `cputool` displays the CPU information, reads and writes of the MSR and the LPC bus.

Tests

There are no defined tests with the `cputool`.

CLI options

```
root@dellemc-diag-os:~# cputool
DellEmc Diag - Cpu Tool
version 1.1, x.xx.x.x-x
build, 2017/05/23,

Syntax: cputool <option>
 Show the help-text:=
      cputool --h                                                              (or)
      cputool -h
 Display the CPU info using CPU-ID:
      cputool --cpuid[=--option]                                               (or)
      cputool -i [option]
 Display the CPU info using x86info:=
      cputool --x86info[=--option]                                             (or)
      cputool -x [option]
 Read CPU register:=
      cputool --readmsr --cpu=<cpuNumber> --reg=<regOffset>                    (or)
      cputool -r -n <cpuNumber> -R <regOffset>
 Write CPU register:=
      cputool --writemsr --cpu=<cpuNumber> --reg=<regOffset> --val=<value>     (or)
      cputool -w <cpuNumber> -R <regOffset> -V <value>
 Execute repeatedly command by count:=
      cputool --iteration=max/<count> [option1] [option2]...                   (or)
      cputool -I max/<count> [option1] [option2]...
 Read the specified regiser in LPC bus:=
      cputool --readlpc --reg=<reg> --size=<size>                              (or)
      cputool -d -R <reg> -Z <size>
 Write the specified regiser in LPC bus:=
      cputool --writelpc --reg=<reg> --val=<value> --size=<size>               (or)
      cputool -W -R <reg> -V <value> -Z <size>

Usage:=
       -h, --h             Show the help text
       -i, --cpuid         CPU-Id
       -x, --x86info       x86 info
       -r, --readmsr       Read operation
       -w, --writemsr      Write operation
       -n, --cpu=          CPU
       -R, --reg=          Register
       -V, --val=          Value to be set
       -Z, --size=         Size
       -I, --iteration=    Iteration command execution
       -d, --readlpc       Read from LPC bus
       -W, --writelpc      Write to LPC bus
```

Output

```
root@dell-diag-os:/# cputool --h
Dell Diag - Cpu Tool
version 1.1, x.xx.x.x
build, 2016/08/12,
Syntax: cputool <option>
 Show the help-text:=
      cputool --h                                                              (or)
      cputool -h
 Display the CPU info using CPU-ID:
      cputool --cpuid[=--option]                                               (or)
      cputool -i [option]
 Display the CPU info using x86info:=
      cputool --x86info[=--option]                                             (or)
      cputool -x [option]
 Read CPU register:=
      cputool --readmsr --cpu=<cpuNumber> --reg=<regOffset>                    (or)
      cputool -r -n <cpuNumber> -R <regOffset>
 Write CPU register:=
      cputool --writemsr --cpu=<cpuNumber> --reg=<regOffset> --val=<value>     (or)
      cputool -w <cpuNumber> -R <regOffset> -V <value>
```

```
 Read the specified regiser in LPC bus:=
      cputool --readlpc --reg=<reg> --size=<size>                                (or)
      cputool -d -R <reg> -Z <size>
 Write the specified regiser in LPC bus:=
      cputool --writelpc --reg=<reg> --val=<value> --size=<size>                 (or)
      cputool -W -R <reg> -V <value> -Z <size>
Usage:=
       -h, --h             Show the help text
       -i, --cpuid         CPU-Id
       -x, --x86info       x86 info
       -r, --readmsr       Read operation
       -w, --writemsr      Write operation
       -n, --cpu=          CPU
       -R, --reg=          Register
       -V, --val=          Value to be set
       -Z, --size=         Size
       -d, --readlpc       Read from LPC bus
       -W, --writelpc      Write to LPC bus
root@dell-diag-os:/#

root@dell-diag-os:/# cputool --x86info
x86info v1.30.  Dave Jones 2001-2011
Feedback to <davej@redhat.com>.
Found 4 identical CPUs
Extended Family: 0 Extended Model: 4 Family: 6 Model: 77 Stepping: 8
Type: 0 (Original OEM)
CPU Model (x86info's best guess): Unknown model.
Processor name string (BIOS programmed): Intel(R) Atom(TM) CPU  C2538  @ 2.40GHz
Total processor threads: 4
This system has 1 dual-core processor with hyper-threading (2 threads per core) running at an
estimated 2.40GHz
root@dell-diag-os:/#
```

eepromtool

To program FRU format EEPROMS, use the `eepromtool`. You can also use the `eepromtool` to show all the FRU-formatted EEPROM contents or show specific EEPROM content by specifying the EEPROM type.

Tests

NOTE: The `eepromtool` tool is used during manufacturing to program FRU data.

CAUTION: The `eepromtool` tool should only be used to read an EEPROM device.

The following command line options are valid cases for running `eepromtool` in Azul.

1. To list the supported eeprom devices type `eepromtool -L`

```
root@dellemc-diag-os:~# eepromtool -L
     MC1EEPROM
     MC2EEPROM
     PSU1EEPROM
     PSU2EEPROM
     FAN1EEPROM
     FAN2EEPROM
     FAN3EEPROM
     FAN4EEPROM
     FAN5EEPROM
     IDEEPROM
root@dellemc-diag-os:~#
```

2. To show a device content type `eepromtool -P <EEPROM_DEVICE> -x`

```
root@dellemc-diag-os:~# eepromtool -P PSU1EEPROM -x
 Board Mfg Date        : Mon Mar 19 03:40:00 2018
Board Mfg             : DELL
Board Product         : PWR SPLY,495W,RDNT,DELTA
 Board Serial          : CNDED0083H0T94
```

```
Board Part Number     : 0GRTNKA02
root@dellemc-diag-os:~#
```

a.

The test option in EEPROM devices allows you to verify the MAC address. Use this test for MAC address consistency.

CLI options

```
DellEmc Diag - Eeprom Tool
version 1.5, x.xx.x.x-x
build, 2017/05/23,

Syntax:= eepromtool <option>
  Display help-text:=
      eepromtool --help                                          (or)
      eepromtool -h
  List the understood TLV codes and names:=
      eepromtool --list                                          (or)
      eepromtool -l
  List all eeprom devices:=
      eepromtool --listdevices                                   (or)
      eepromtool -L
  Dump the PSU eeprom:=
      eepromtool --psueepromdump                                 (or)
      eepromtool -m
  Dump the FAN eeprom:=
      eepromtool --faneepromdump                                 (or)
      eepromtool -F
  Show the EEPROM data:=
     eepromtool --eeprom=<eepromtype> --show                     (or)
     eepromtool -P <eepromtype>  -x
  Reset the EEPROM data:=
     eepromtool --eeprom=<eepromtype> --erase                    (or)
     eepromtool -P <eepromtype>  -e
  Verify the MAC address in system-eeprom and mac-eeprom:=
     eepromtool --eeprom=<eepromtype> --test                     (or)
     eepromtool -P <eepromtype>  -t
  Look up a TLV by code and write the value to stdout:=
     eepromtool --eeprom=<eepromtype> --get <code>               (or)
     eepromtool -P <eepromtype>  -g <code>
  Execute repeatedly command by count:=
     eepromtool --iteration=max/<count> [option1] [option2]...  (or)
     eepromtool -I max/<count> [option1] [option2]...
  Set a TLV code to a value:=
     eepromtool --eeprom=<eepromtype> --set <code>=<value>,<code>=<value>...(or)
     eepromtool -P <eepromtype> -s <code>=<value>,<code>=<value>...

Usage:=
    -h, --h              Show the help text
    -l, --list           List the understood TLV codes and names
    -L, --listdevices    List all EEPROM devices
    -m, --psueepromdump  Dump the PSU EEPROM
    -F, --faneepromdump  Dump the FAN EEPROM
    -P, --eeprom=        EEPROM type
    -x, --show           Show operation
    -e, --erase          Erase operation
    -t, --test           Test using the pre-programmed configuration or use supplied config
    -I, --iteration=     Iteration command execution
    -g, --get            Get operation
    -s, --set            Set operation
```

Output

```
root@dell-diag-os:/opt/ngos/bin# eepromtool --list
TLV Code TLV Name
======== =================
0x21 Product Name
```

```
0x22 Part Number
0x23 Serial Number
0x24 Base MAC Address
0x25 Manufacture Date
0x26 Device Version
0x27 Label Revision
0x28 Platform Name
0x29 Loader Version
0x2a MAC Addresses
0x2b Manufacturer
0x2c Country Code
0x2d Vendor Name
0x2e Diag Version
0x2f Service Tag
0xfd Vendor Extension
0xfe CRC-32
root@dell-diag-os:/opt/ngos/bin# eepromtool --listdevices
CPUEEPROM1
CPUEEPROM2
CPUEEPROM3
CPUEEPROM4
CPUEEPROM5
CPUEEPROM6
CPUEEPROM7
CPUEEPROM8
FAN1EEPROM
FAN2EEPROM
FAN3EEPROM
FAN4EEPROM
FAN5EEPROM
SwitchEEPROM
root@dell-diag-os:/# eepromtool --psueepromdump
************PSU1_CountryCode*************
Registers 0x24a - 0x24b
CN
************PSU1_DellPartNumber*************
Registers 0x24c - 0x251
02RPHX
************PSU1_MfgID*************
Registers 0x252 - 0x256
17972
************PSU1_MfgDate*************
Registers 0x257 - 0x25e
151117
************PSU1_SerialNo*************
Registers 0x25f - 0x262
01CG
************PSU1_ServiceTag*************
Registers 0x263 - 0x269
************PSU1_LabelRevision*************
Registers 0x26a - 0x26c
A00
************PSU2_CountryCode*************
Registers 0x283 - 0x284
CN
************PSU2_DellPartNumber*************
Registers 0x285 - 0x28a
02RPHX
************PSU2_MfgID*************
Registers 0x28b - 0x28f
17972
************PSU2_MfgDate*************
Registers 0x290 - 0x297
151117
************PSU2_SerialNo*************
Registers 0x298 - 0x29b
015F
************PSU2_ServiceTag*************
Registers 0x29c - 0x2a2
************PSU2_LabelRevision*************
Registers 0x2a3 - 0x2a5
A00
root@dell-diag-os:/#
```

```
root@dell-diag-os:/opt/ngos/bin#
root@dell-diag-os:/opt/ngos/bin# eepromtool --eeprom=cpueeprom2 --set 0x21='cpu2'
Notice: Invalid TLV checksum found. Using default contents.
Adding TLV 0x21: Product Name
Programming passed.
TlvInfo Header:
Id String: TlvInfo
Version: 1
Total Length: 12
TLV Name Code Len Value
-------------------- ---- --- -----
Product Name 0x21 4 cpu2
CRC-32 0xFE 4 0x338B2B86
Checksum is valid.
root@dell-diag-os:/opt/ngos/bin#
root@dell-diag-os:/opt/ngos/bin# eepromtool --eeprom=cpueeprom2 --get 0x21
cpu2
root@dell-diag-os:/opt/ngos/bin#
root@dell-diag-os:/opt/ngos/bin# eepromtool --eeprom=cpueeprom2 --show
TlvInfo Header:
Id String: TlvInfo
Version: 1
Total Length: 12
TLV Name Code Len Value
-------------------- ---- --- -----
Product Name 0x21 4 cpu2
CRC-32 0xFE 4 0x338B2B86
Checksum is valid.
root@dell-diag-os:/opt/ngos/bin#
root@dell-diag-os:/opt/ngos/bin# eepromtool --eeprom=cpueeprom1 --erase
Programming passed.
EEPROM does not contain data in a valid TlvInfo format.
root@dell-diag-os:/opt/ngos/bin# eepromtool --eeprom=cpueeprom1 --show
Notice: Invalid TLV header found. Using default contents.
Notice: Invalid TLV checksum found. Using default contents.
TlvInfo Header:
Id String: TlvInfo
Version: 1
Total Length: 6
TLV Name Code Len Value
-------------------- ---- --- -----
CRC-32 0xFE 4 0xD4431C18
Checksum is valid.
root@dell-diag-os:/opt/ngos/bin#
```

ethtool

To control and query network drivers and hardware use the `ethtool`.

Tests

root@dellemc-diag-os:/opt/dellemc/diag/bin# ethtool -t ethx

CLI options

```
ethtool -h|--help
ethtool --version
ethtool -a|--show-pause devname
ethtool -A|--pause devname [autoneg on|off] [rx on|off] [tx on|off]
ethtool -c|--show-coalesce devname
ethtool -C|--coalesce devname [adaptive-rx on|off] [adaptive-tx on|off] [rx-usecs N] [rx-
frames N] [rx-usecs-irq N] [rx-frames-irq N] [tx-usecs N] [tx-frames N] [tx-usecs-irq N] [tx-
frames-irq N] [stats-block-usecs N] [pkt-rate-low N] [rx-usecs-low N] [rx-frames-low N] [tx-
usecs-low N] [tx-frames-low N] [pkt-rate-high N] [rx-usecs-high N] [rx-frames-high N] [tx-
usecs-high N] [tx-frames-high N] [sample-interval N]
ethtool -g|--show-ring devname
```

```
ethtool -G|--set-ring devname [rx N] [rx-mini N] [rx-jumbo N] [tx N]
ethtool -i|--driver devname
ethtool -d|--register-dump devname [raw on|off] [hex on|off] [file name]
ethtool -e|--eeprom-dump devname [raw on|off] [offset N] [length N]
ethtool -E|--change-eeprom devname [magic N] [offset N] [length N] [value N]
ethtool -k|--show-features|--show-offload devname
ethtool -K|--features|--offload devname feature on|off ...
ethtool -p|--identify devname [N]
ethtool -P|--show-permaddr devname
ethtool -r|--negotiate devname
ethtool -S|--statistics devname
ethtool -t|--test devname [offline|online|external_lb]
ethtool -s devname speed N [duplex half|full] [port tp|aui|bnc|mii] [autoneg on|off]
[advertise N] [phyad N] [xcvr internal|external] [wol p|u|m|b|a|g|s|d...] [sopass
xx:yy:zz:aa:bb:cc] [msglvl N | msglvl type on|off ...]
ethtool -n|-u|--show-nfc|--show-ntuple devname [ rx-flow-hash tcp4|udp4|ah4|esp4|sctp4|tcp6|
udp6|ah6|esp6|sctp6 |
rule N ]
ethtool -N|-U|--config-nfc|--config-ntuple devname rx-flow-hash tcp4|udp4|ah4|esp4|sctp4|tcp6|
udp6|ah6|esp6|sctp6 m|v|t|s|d|f|n|r... |
flow-type ether|ip4|tcp4|udp4|sctp4|ah4|esp4 [src xx:yy:zz:aa:bb:cc [m xx:yy:zz:aa:bb:cc]]
[dst xx:yy:zz:aa:bb:cc [m xx:yy:zz:aa:bb:cc]] [proto N [m N]] [src-ip x.x.x.x [m x.x.x.x]]
[dst-ip x.x.x.x [m x.x.x.x]] [tos N [m N]] [l4proto N [m N]] [src-port N [m N]] [dst-port N
[m N]] [spi N [m N]] [l4data N [m N]] [vlan-etype N [m N]] [vlan N [m N]] [user-def N [m N]]
[action N] [loc N] |
delete N
ethtool -w|--get-dump devname [data filename]
ethtool -W|--set-dump devname N
ethtool -T|--show-time-stamping devname
ethtool -x|--show-rxfh-indir devname
ethtool -X|--set-rxfh-indir devname [ equal N | weight W0 W1 ... ]
ethtool -f|--flash devname FILE [N]
ethtool -l|--show-channels devname
ethtool -L|--set-channels devname [rx N] [tx N] [other N] [combined N]
ethtool -m|--dump-module-eeprom devname [raw on|off] [hex on|off] [offset N] [length N]
ethtool --show-priv-flags devname
ethtool --set-priv-flags devname flag on|off ...
ethtool --show-eee devname
ethtool --set-eee devname [eee on|off] [tx-lpi on|off] [tx-timer N] [advertise N]
```

Output

```
The test result is PASS
The test extra info:
Register test  (offline)         0
Eeprom test    (offline)         0
Interrupt test (offline)         0
Loopback test  (offline)         0
Link test   (on/offline)         0
```

fantool

The `fantool` tests the fans in the system, sets and reports the fan speeds and the fan tray field replaceable unit (FRU) registers.

The `fantool` also reports the airflow direction of the fans. The `psutool` command controls the PSU fans.

Tests

The `fantool` tests the fans by setting them to different speeds and then verifying the configured fan speeds.

Registers and values pass as hexadecimal values with or without the preceding `0x`. Fans display from `1` to `Max System Fans`.

CLI options

```
DellEmc Diag - Fan Controller Tool
version 1.5, x.xx.x.x-x
build, 2017/05/23

Syntax: fantool <option>
 Show the help-text:=
     fantool --h                                                  (or)
     fantool -h
 Initialize the fans to the default state:=
     fantool --init                                               (or)
     fantool -i
 Test using the Fan Controller config file:=
     fantool --test [--fan=<fan>] [--lpc]                         (or)
     fantool -t [-F <fan>] [-l]
 Get the speed of the specified fan or all fans in RPM:=
     fantool --get --fan=<fan | all> [--lpc]                      (or)
     fantool -g -F <fan | all> [-l]
 Set the fan(s) to the speed:=
     fantool --set --fan=<fan | all> --speed=<speed in RPM>       (or)
     fantool -s -F <fan | all> -S <speed in RPM>
 Execute repeatedly command by count:=
     fantool --iteration=max/<count> [option1] [option2]...       (or)
     fantool -I max/<count> [option1] [option2]...
 Read the Register from the fan controller:=
     fantool --read --fan=<fan | all> --reg=<register | all>      (or)
     fantool -r -F <fan | all> -R <register | all>
 Write the Register in the Fan Controller:=
     fantool --write --fan=<fan | all> --reg=<register> --val=<value> (or)
     fantool -w -F <fan | all> -R <register> -V <value>

Usage:=
     -h, --h             Show the help text
     -i, --init          Initilize to default
     -t, --test          Test using the pre-programmed configuration or use supplied config
     -g, --get           Get operation
     -s, --set           Set operation
     -r, --read          Read operation
     -w, --write         Write operation
     -I, --iteration=    Iteration command execution
     -F, --fan=          Fan Id
     -R, --register=     Register
     -V, --val=          Value to be set
     -S, --speed=        Speed of the fan
     -q, --lpc           Test by reading or modifying SmartFusion registers.
                         When this flag is used, it must be clubbed with one of above flags

  *Registers and Values are passed as Hexadecimal values with or without the preceding 0x.
  *Fans are from 1 to Max System Fans.
```

The `fantool` uses long options which requires two hyphens in front of the options. Options are required, optional, or none. If you require a parameter, specify it and include an equal sign. If a parameter is optional, enclose it with square brackets to show that it is optional, but do not type the brackets at the CLI. For example, `--fan` is optional and enter it as `--fan=1` or `--fan=all`, and so forth. Parameters with angle brackets are required but have multiple options for the input. Do not type the angle brackets or the vertical line character in the CLI. Only use one option per command; for example, `--fan=1` or `--fan=all`.

- `test` — Runs through the speeds for the fan, from highest to lowest, and checks that the fan can run at the speeds of the test. If a single fan is listed on the CLI, that fan is tested. If you use the `all` option, all fans are tested. The number in the parentheses during the test is the speed the system tries to reach during the test. If a fan cannot reach the desired speed within an acceptable range after 10 checks, the fan fails for that speed and the system moves on to the next fan.
- `get` — Gets the speed of the fan and returns it in the rate process module (RPM).
- `set` — Sets the speed of the fan in the RPM.

NOTE: Commonly, fan speeds are in two registers and must be written in a specific order. The `write` command cannot change the fan speeds; use the `set` command.

Output

test output

```
root@dell-diag-os:~# fantool --test --lpc
Fan Controller Test LPC.................................
Max number of Fan Trays in the System : 5
Number of fans per tray : 2
Max Fan Speed set(PWM): 255
Getting Details for  Fan 1
 Fan 1 is Present
 Fan 1 Air flow type is Front To Rear
 Fan 1 status Normal
 Fan 1 speed is 8420 RPM
Getting Details for  Fan 2
 Fan 2 is Present
 Fan 2 Air flow type is Front To Rear
 Fan 2 status Normal
 Fan 2 speed is 8738 RPM
Getting Details for  Fan 3
 Fan 3 is Present
 Fan 3 Air flow type is Front To Rear
 Fan 3 status Normal
 Fan 3 speed is 8474 RPM
Getting Details for  Fan 4
 Fan 4 is Present
 Fan 4 Air flow type is Front To Rear
 Fan 4 status Normal
 Fan 4 speed is 8757 RPM
Getting Details for  Fan 5
 Fan 5 is Present
 Fan 5 Air flow type is Front To Rear
 Fan 5 status Normal
 Fan 5 speed is 8492 RPM
Getting Details for  Fan 6
 Fan 6 is Present
 Fan 6 Air flow type is Front To Rear
 Fan 6 status Normal
 Fan 6 speed is 8777 RPM
Getting Details for  Fan 7
 Fan 7 is Present
 Fan 7 Air flow type is Front To Rear
 Fan 7 status Normal
 Fan 7 speed is 8348 RPM
Getting Details for  Fan 8
 Fan 8 is Present
 Fan 8 Air flow type is Front To Rear
 Fan 8 status Normal
 Fan 8 speed is 8585 RPM
Getting Details for  Fan 9
 Fan 9 is Present
 Fan 9 Air flow type is Front To Rear
 Fan 9 status Normal
 Fan 9 speed is 8420 RPM
Getting Details for  Fan 10
 Fan 10 is Present
 Fan 10 Air flow type is Front To Rear
 Fan 10 status Normal
 Fan 10 speed is 8566 RPM
Fan Controller Test LPC..................... Passed
root@dell-diag-os:~#

root@dell-diag-os:~# fantool --get --lpc
Fan 1 speed is 8420 RPM
Fan 2 speed is 8757 RPM
Fan 3 speed is 8474 RPM
Fan 4 speed is 8738 RPM
Fan 5 speed is 8474 RPM
Fan 6 speed is 8757 RPM
Fan 7 speed is 8366 RPM
```

```
Fan 8 speed is 8604 RPM
Fan 9 speed is 8420 RPM
Fan 10 speed is 8566 RPM
[2]+  Done                    dhclient -q eth0
root@dell-diag-os:~#

root@dell-diag-os:~# fantool --get --fan=2 --lpc
Fan 2 speed is 8738 RPM
root@dell-diag-os:~#
```

flashrom

To update or erase the BIOS flash memory, the `smbiostool` uses `flashrom`.

gpiotool

The `gpiotool` controls the state of the GPIO lines from the CPU or any other device that drives the GPIO lines.

The CPU GPIO alines the map in Linux to `/sys/class/gpio` entries, which are manipulated through the standard read/write interfaces. There is chip numbering to support multiple GPIO chips, or chips at an offset. For devices such as the complex programmable logic device (CPLD) or field programmable gate arrays (FPGA), `gpiotool` accesses those devices to drive the GPIO lines using the standard bus interfaces such as `i2c`, `mem`, or `pci`.

CLI options

```
DellEmc Diag - GPIO Tool
version 1.4, x.xx.x.x-x
build, 2017/05/23,

Syntax: gpiotool <option>
 Show the help-text:=
      gpiotool --h                                                (or)
      gpiotool -h
 List available gpio chips and pins:=
      gpiotool --list                                             (or)
      gpiotool -l
 Set GPIO pin:=
      gpiotool --set [--chip=<chip>] --pin=<pin> --val=<value> (or)
      gpiotool -s [-c <chip>] -H <pin> -V <value>
 Get GPIO pins value:=
      gpiotool --get [--chip=<chip>] [--pin=<pin>]             (or)
      gpiotool -g [-c <chip>] [-H <pin>]
 Execute repeatedly command by count:=
      gpiotool --iteration=max/<count> [option1] [option2]...  (or)
      gpiotool -I max/<count> [option1] [option2]...
Usage:=
      -h, --h           Show the help text
      -l, --list        List the understood TLV codes and names
      -s, --set         Set operation
      -g, --get         Get operation
      -c, --chip=       GPIO chip
      -I, --iteration=  Iteration command execution
      -H, --pin=        GPIO pin number
      -V, --val=        Value to be set
```

Output

list output

```
root@dell-diag-os:~# gpiotool --list
Chip 0  Core Gpio  bits: 60 CORE gpiochip196
```

```
=======================================
Bit             Name     Dir  AC  Value
=======================================
15         SATA_GP0       IN   LOW    0
16         SATA_LEDN      OUT  LOW    0
17         SATA3_GP0      IN   LOW    0
19         FLEX_CLK_SE0   IN   LOW    0
20         FLEX_CLK_SE1   IN   LOW    0
32            GPIO_SUS1   IN   LOW    0
33            GPIO_SUS2   OUT  LOW    0
34          CPU_RESET_B   OUT  LOW    0
36           PMU_SUSCLK   OUT  LOW    0
37    PMU_SLP_DDRVTT_B    IN   LOW    0
38       PMU_SLP_LAN_B    IN   LOW    0
39          PMU_WAKE_B    OUT  LOW    0
40        PMU_PWRBTN_B    IN   LOW    0
49          GBE_SDP0_1    IN   LOW    0
50            GBE_LED0    IN   LOW    0
51            GBE_LED1    IN   LOW    0
52            GBE_LED2    IN   LOW    0
53            GBE_LED3    IN   LOW    0
54           NCSI_RXD1    OUT  LOW    0
55   GBE_MDIO0_I2C_CLK    OUT  LOW    0
58  GBE_MDIO1_I2C_DATA    IN   LOW    0
59           JTAG_TRST    OUT  LOW    0
root@dell-diag-os:~#
```

get output

```
root@dell-diag-os:~# gpiotool --get --pin=1
Chip 0  Core Gpio  bits: 60 CORE gpiochip196
===================================
Bit             Name     Dir  Value
===================================
```

set output

```
root@amazon:/opt/ngos/bin# ./gpiotool --set --pin=1 --val=1
```

i2ctool

The `i2ctool` allows for scanning, reading, and writing of the `I2c` bus devices.

To read and write to devices on the `i2c` bus, use the `i2ctool`. The `i2ctool` also scans the `i2c` busses and reports what devices are found. The scan reads address 0x0 from all the devices in the address range of 0x0 to 0x7f on all `i2c` busses present. The `i2ctool` does not automatically traverse MUXes along the `i2c` bus. Other tools use this tool to read `i2c` device information and pass the results back through a named pipe.

Tests

To test, the `i2ctool` has a configuration file that lists all the devices on the busses. The tool runs through the list and tries to reach the devices. The `i2ctool` reports when a device is not returning data.

CLI options

```
DellEmc Diag - I2C Tool
version 1.5, x.xx.x.x-x
build, 2017/05/23,

Syntax: i2ctool <option>
```

```
 To Scan the (Specific) I2C devices:=
        i2ctool --scan [--bus=/dev/i2c-
<bus_number>]
                    (or)
        i2ctool -n [-b /dev/i2c-<bus_number>]
 To Test the pre-programmed configuration or from config file:=
        i2ctool --test [--
config=<config_file_name>]
                                (or)
        i2ctool -t [-f <config_file_name>]
 Execute repeatedly command by count:=
        i2ctool --iteration=max/<count> [option1]
[option2]...
         (or)
        i2ctool -I max/<count> [option1] [option2]...
 Read:=
        i2ctool --read --bus=/dev/i2c-<bus_number> --addr=<address> --reg=<register> --
count=<count> --width=<width> --display_size=<display_size>          (or)
        i2ctool -r -b /dev/i2c-<bus_number> -a <address> -R <register> -C <count> -W <width> -
D <display_size>
 Read(16 bit addressing):=
        i2ctool --read --bus=/dev/i2c-<bus_number> --addr=<address> --reg16=<register(16bit)>
[--reg_le] --count=<count> --width=<width> --display_size=<display_size> (or)
        i2ctool -r -b /dev/i2c-<bus_number> -a <address> -o <register(16bit)> [-L] -C <count>
-W <width> -D <display_size>
 Write:=
        i2ctool --write --bus=/dev/i2c-<bus_number> --addr=<address> --reg=<regiser> --
width=<width>  --val=<value>                                    (or)
        i2ctool -w -b /dev/i2c-<bus_number> -a <address> -R <register> -W <width> -V <value>
 Write(16 bit addressing):=
        i2ctool --write --bus=/dev/i2c-<bus_number> --addr=<address> --
reg16=<register(16bit)> [--reg_le] --val=<value>
(or)
        i2ctool -w -b /dev/i2c-<bus_number> -a <address> -o <register(16bit)> [-L] -V <value>

Usage:
  -h, --h              Show the help text
  -n, --scan           Scan operation
  -t, --test           Test using the pre-programmed configuration or use supplied config
  -r, --read           Read operation
  -w, --write          Write operation
  -f, --config=        To specify the location of the config file e.g. /etc/dn/diag/
<file_name>
  -C, --count=         Count
  -R, --reg=           Register
  -o, --reg16=         Register(16 bit addressing)
  -V, --val=           Value to be set
  -W, --width=         Width {8,16}
  -b, --buspath=       To specify the i2c bus   e.g.: /dev/i2c-<bus number>
  -a, --addr=          Address
  -D, --display_size=  Display size, {1,2,4} of bytesDisplay size, {1,2,4} of bytes
  -I, --iteration=     Iteration command execution
```

Output

NOTE: The `i2ctool` does not automatically scan multiple MUXed segments. Before scanning, you MUST set the MUXes to select the devices you want to see on the busses. By default, the `i2ctool` scans the `i2c` devices from the root MUX where it sees the list of devices directly connected to the CPU MUX. The default scan function scans all connected busses. By specifying a bus, you can limit the scan to one bus. In the scan data, `RR` indicates a reserved address which is not used for any devices and `UU` indicates that the device is busy or mapped to the OS.

`scan` Output

```
root@dell-diag-os:/etc/dn/diag# i2ctool --scan
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: RR RR RR RR RR RR RR RR -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- 18 -- 1a -- -- -- -- --
```

```
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 2e --
30: 30 -- 32 -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 -- 52 -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 69 -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- RR RR RR RR RR RR RR RR
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: RR RR RR RR RR RR RR RR -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 3e --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: 50 51 52 53 54 55 56 57 -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- -- RR RR RR RR RR RR RR RR
I2C devices found on bus #0: 8
 0x18  0x1a  0x2e  0x30  0x32  0x50  0x52  0x69
I2C devices found on bus #1: 10
 0x3e  0x50  0x51  0x52  0x53  0x54  0x55  0x56
 0x57  0x70
root@dell-diag-os:/etc/dn/diag#
```

test Output

```
root@dell-diag-os:/etc/dn/diag# i2ctool --test
Testing I2C devices:
Checking I2C devices on bus 0:
+ Checking Clock GEN                0x69  ..... Passed
+ Checking SPD0                     0x50  ..... Passed
Checking I2C devices on bus 1:
+ Checking CPU Board I2C Mux        0x70  ..... Passed
+ Checking CPU Board EEPROM1        0x53  ..... Passed
+ Checking CPU Board EEPROM2        0x57  ..... Passed
+ Checking Switch Brd EEPROM        0x50  ..... Passed
+ Checking CPLD2                    0x3e  ..... Passed
+ Checking CPLD3                    0x3e  ..... Passed
+ Checking CPLD4                    0x3e  ..... Passed
+ Checking SFP+ 1                   0x50  ..... Passed
+ Checking SFP+ 2                   0x50  ..... Passed
+ Checking SFP+ 3                   0x50  ..... Passed
+ Checking SFP+ 4                   0x50  ..... Passed
+ Checking SFP+ 5                   0x50  ..... Passed
+ Checking SFP+ 6                   0x50  ..... Passed
+ Checking SFP+ 7                   0x50  ..... Passed
+ Checking SFP+ 8                   0x50  ..... Passed
+ Checking SFP+ 9                   0x50  ..... Passed
+ Checking SFP+ 10                  0x50  ..... Passed
+ Checking SFP+ 11                  0x50  ..... Passed
+ Checking SFP+ 12                  0x50  ..... Passed
+ Checking SFP+ 13                  0x50  ..... Passed
+ Checking SFP+ 14                  0x50  ..... Passed
+ Checking SFP+ 15                  0x50  ..... Passed
+ Checking SFP+ 16                  0x50  ..... Passed
+ Checking SFP+ 17                  0x50  ..... Passed
+ Checking SFP+ 18                  0x50  ..... Passed
+ Checking SFP+ 19                  0x50  ..... Passed
+ Checking SFP+ 20                  0x50  ..... Passed
+ Checking SFP+ 21                  0x50  ..... Passed
+ Checking SFP+ 22                  0x50  ..... Passed
+ Checking SFP+ 23                  0x50  ..... Passed
+ Checking SFP+ 24                  0x50  ..... Passed
+ Checking SFP+ 25                  0x50  ..... Passed
+ Checking SFP+ 26                  0x50  ..... Passed
+ Checking SFP+ 27                  0x50  ..... Passed
+ Checking SFP+ 28                  0x50  ..... Passed
+ Checking SFP+ 29                  0x50  ..... Passed
+ Checking SFP+ 30                  0x50  ..... Passed
+ Checking SFP+ 31                  0x50  ..... Passed
+ Checking SFP+ 32                  0x50  ..... Passed
+ Checking SFP+ 33                  0x50  ..... Passed
```

```
+ Checking SFP+ 34                   0x50   ..... Passed
+ Checking SFP+ 35                   0x50   ..... Passed
+ Checking SFP+ 36                   0x50   ..... Passed
+ Checking SFP+ 37                   0x50   ..... Passed
+ Checking SFP+ 38                   0x50   ..... Passed
+ Checking SFP+ 39                   0x50   ..... Passed
+ Checking SFP+ 40                   0x50   ..... Passed
+ Checking QSFP+ 41                  0x50   ..... Passed
+ Checking QSFP+ 42                  0x50   ..... Passed
+ Checking QSFP28 43                 0x50   ..... Passed
+ Checking QSFP28 44                 0x50   ..... Passed
+ Checking QSFP28 45                 0x50   ..... Passed
+ Checking QSFP28 46                 0x50   ..... Passed
+ Checking QSFP28 47                 0x50   ..... Passed
+ Checking QSFP28 48                 0x50   ..... Passed
I2C Devices: Overall test results -------- >>> Passed
root@dell-diag-os:/etc/dn/diag#
```

read Output

```
/opt/ngos/bin# ./i2ctool --read --bus=/dev/i2c-1 --addr=0x50 --reg=0 --count=256
0x92 0x13 0x0b 0x08 0x04 0x21 0x02 0x09 0x0b 0x11 0x01 0x08 0x0c 0x00 0x7e 0x00
0x69 0x78 0x69 0x30 0x69 0x11 0x20 0x89 0x20 0x08 0x3c 0x3c 0x00 0xf0 0x83 0x05
0x80 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x85 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0f 0x11 0x23 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x80 0x2c 0x0f 0x13 0x35 0xe9 0x8d 0xe0 0xbb 0x80 0x50
0x31 0x38 0x4b 0x53 0x46 0x31 0x47 0x37 0x32 0x48 0x5a 0x2d 0x31 0x47 0x34 0x45
0x32 0x20 0x45 0x32 0x80 0x2c 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
```

write Output

```
/opt/dell/diag/bin# ./i2ctool --write --bus=/dev/i2c-2 --addr=0x48 --reg=0x14 --val=1
```

ledtool

The `ledtool` allows you to control the state of the front and back panel light emitting diodes (LEDs). ASIC and Phys control the port LEDs and are beyond the scope of this tool.

You can manually control the front and back panel LEDs normally controlled through the CPLD or FPGA access. When set, bits in these registers control the state of the LED.

Tests

To test the LEDs, use the `ledtool --test` command.

```
root@dell-diag-os:/opt/ngos/bin# ./ledtool --test
LED Test Started... Will take few mins to complete.
LED Tool: Overall test results --------------------- >>> Passed
```

CLI options

```
DellEmc Diag - Led Tool
version 1.0, x.xx.x.x-x
build, 2017/05/23,

Usage:
 List the LEDs:=
     ledtool --list                                                                    (or)
     ledtool -l
 Get the state of (specific) LED(s):=
     ledtool --get [--led=<led>]                                                       (or)
     ledtool -g [-D <led>]
 Set the state of specific LED(color and blink):=
     ledtool --set --led=<led> [--instance=<instance>] {--state=<state> | --val=<value>} (or)
     ledtool -s -D <led> [-I <instance>] {-T <state> | -V <value>}
 Execute repeatedly command by count:=
     ledtool --iteration=max/<count> [option1] [option2]...                            (or)
     ledtool -I max/<count> [option1] [option2]...
 Test using config file:=
     ledtool --test [--config=<config_file>]                                           (or)
     ledtool -t [-f <config_file>]

Syntax: ledtool <option>
  -h, --h            Show the help text
  -l, --list         List the LEDs
  -g, --get          Get operation
  -s, --set          Set operation
  -t, --test         Test using the pre-programmed configuration or use supplied config
  -D, --led=         LED
  -I, --iteration=   Iteration command execution
  -S, --instance=,   Instance
  -T, --state=,      State of the LED
  -V, --val=,        Value to be set
  -f, --config=,     To specify the location of the config file e.g. /etc/dn/diag/<file_name>

[led] selections are:
Power
States: green amber flashing-amber off
System
States: amber flashing-green flashing-amber green
Fan
States: green flashing-amber off
Beacon
States: flashing-blue off
CPLD2-Mode
States: normal-mode test-mode
Port#1-18-Amber
States: off flashing-amber-fast amber flashing-amber
Port#1-18-Green
States: off flashing-green-fast green flashing-green
CPLD3-Mode
States: normal-mode test-mode
Port#19-36-Amber
States: off flashing-amber-fast amber flashing-amber
Port#19-36-Green
States: off flashing-green-fast green flashing-green
CPLD4-Mode
States: normal-mode test-mode
Port#37-48-Amber
States: off flashing-amber-fast amber flashing-amber
Port#37-48-Green
States: off flashing-green-fast green flashing-green
```

Output

list output

```
root@dell-diag-os:/etc/dn/diag# ledtool --list
    Power Led  : options
        green amber flashing-amber off
    System Led  : options
        amber flashing-green flashing-amber green
    Fan Led  : options
        green flashing-amber off
    Beacon LED  : options
        flashing-blue off
    Ports 1-18 PortLED Mode        : options
        normal-mode test-mode
    Ports 1-18 FrontEnd AmberLed   : options
        off flashing-amber-fast amber flashing-amber
    Ports 1-18 FrontEnd GreenLed   : options
        off flashing-green-fast green flashing-green
    Ports 19-36 PortLED Mode       : options
        normal-mode test-mode
    Ports 19-36 FrontEnd AmberLed  : options
        off flashing-amber-fast amber flashing-amber
    Ports 19-36 FrontEnd GreenLed  : options
        off flashing-green-fast green flashing-green
    Ports 37-48 PortLED Mode       : options
        normal-mode test-mode
    Ports 37-48 FrontEnd AmberLed  : options
        off flashing-amber-fast amber flashing-amber
    Ports 37-48 FrontEnd GreenLed  : options
        off flashing-green-fast green flashing-green
root@dell-diag-os:/etc/dn/diag#
```

get Output

```
root@dell-diag-os:/etc/dn/diag# ledtool --get
    Power Led : flashing-amber
    System Led : flashing-green
    Fan Led : green
    Beacon LED : off
    Ports 1-18 PortLED Mode       : normal-mode
    Ports 1-18 FrontEnd AmberLed  : off
    Ports 1-18 FrontEnd GreenLed  : off
    Ports 19-36 PortLED Mode      : normal-mode
    Ports 19-36 FrontEnd AmberLed : off
    Ports 19-36 FrontEnd GreenLed : off
    Ports 37-48 PortLED Mode : normal-mode
    Ports 37-48 FrontEnd AmberLed : off
    Ports 37-48 FrontEnd GreenLed : off
root@dell-diag-os:/etc/dn/diag#
```

lpctool

To access devices on the LPC bus, use the `lpctool`.

The `lpctool` allow access on the LPC bus by using I/O transactions at the processor level. This access does not include LPC interfaces in other devices. Other DiagOS tools use `lpctool` to read LPC-connected registers.

CLI options

```
DellEmc Diag - LPC Tool
version 1.0, x.xx.x.x-x
build, 2017/05/23,

Syntax: lpctool <option>
   Show the help-text:=
       lpctool --h                                                                    (or)
       lpctool -h
   Read the specified address:=
       lpctool --read --addr=<address> --count=<number_of_bytes> [--size=<b,w or l>] (or)
       lpctool -r -a <address> -C <number_of_bytes> [-z <b,w or l>]
   Write data at the specified address:=
       lpctool --write --addr=address --val=data [--size=b,w or l]                    (or)
       lpctool -w -a <address> -V <data> [-z <b,w or l>]
   Execute repeatedly command by count:=
       lpctool --iteration=max/<count> [option1] [option2]...                         (or)
       lpctool -I max/<count> [option1] [option2]...
Usage:=
       -h, --h           Show the help text
       -w, --write       Write operation
       -r, --read        Read operation
       -z, --size=       Size
       -I, --iteration=  Iteration command execution
       -C, --count=      Count
       -a, --addr=       Address
```

Output

read Output

```
root@dell-diag-os:/opt/ngos/bin# ./lpctool --read --addr=102
Byte Port 0x102 : 0xde
```

write Output

```
root@dell-diag-os:/opt/ngos/bin# ./lpctool --write --addr=102 --val=10
```

memtool

The `memtool` tests the physical memories in the system.

The `memtool` performs address bus and data tests that moves 1s or 0s through the bus lines to detect stuck, missing, bridged, or other issues found during board tests. The tool also places hamming values or addresses into memory to test and report failing bits. All tests are similar to the `memtest86` application but are available through the CLI.

In addition, the `memtool` reads the types and locations of memory in the system. The memory may be physical RAMs connected to the CPU covered by caches, or memory attached or embedded in other devices or across buses. The tool must know the addressable location of the memory, the memory address, data bus sizes, and any addressing constraints; for example, byte or word addressable boundaries.

The `memtool` allocates a memory region to tests in, which is either `malloc` space or opens a memory map to the memory, and passes the pointer to access the memory.

Tests

- `Address Read`—Causes read transactions on the memory bus. `Address read` can loop for several iterations, checking for any changes in the data between iterations. You can specify patterns on the address bus for the bits to allow the testing for stuck address bits.

- `Address Write`—Creates write transactions on the memory bus. `Address write` can loop for several iterations, and works similar to the `Address Read` test.
- `Address Walking 1`—Walks a 1 though the provided address space in memory for the available address bits. `Address Walking 1` writes the address of the cell in the location it is referencing. After it is done writing all the locations, it walks back through and verifies that the data is correct.
- `Address Walking 0`—Walks a 0 address bit through the memory area available to it. `Address walking 0` writes the additive inverse of the address to the location. After writing all addressed locations, it walks back through and verifies the locations data.
- `Data Read`—Reads transactions similar to the `Address Read` test, but focuses on the data bits. Patterns are placed on the data bus to test for stuck data bits.
- `Data Write`—Places data patterns on the bus for testing the bus and looks for stuck data bits.
- `Data Walking 1`—Walks a 1 through the data bits within an address location and verifies that the values are valid before overwriting.
- `Data Walking 0`—Walks a 0 through the data bits and verifies the value as it is testing.
- `Data Sliding 1`—Slides a 1 through the data testing for stuck bits. By `xor` of each shift to the data, when finished, the cell holds all the 1s.
- `Data Sliding 0`—Slides a 0 through the data bits set to 1. By `xor` of each shift of the data, when finished, the cell holds all the 1s.
- `Data Pattern`—Writes different patterns to memory locations within the specified region. The patterns are 0xFFFF, 0xFF00, 0xF0F0, 0xAAAA, 0xAA55 and 0x5555. The patterns are written as repeated portions of these patterns in the memory to fill the memory and as Hamming patterns (such as Hamming [8,4], Hamming[16,11], Hamming[32,26] or Hamming[64,57]) encoding with the additional most significant byte (MSB) parity bit to cover the parity bits in the Hamming code. This pattern allows for detecting multiple bit errors.
- `Data Cache`—Performs a rotation of a 16MB array in four clockwise rotations for 16 iterations of the complete rotation. The 16MB size ensures that memory is not within the cache lines and causes cache ejections through each of the rotations.

CLI options

```
DellEmc Diag - Memory Tool
version 1.5, x.xx.x.x-x
build, 2017/05/23,

Syntax: memtool <option>
 Show the Help-text:=
      memtool --h                                                                    (or)
      memtool -h
 Display the configuration info of the device:=
      memtool --info                                                                 (or)
      memtool -i
 List all of the memory regions in the config file:=
      memtool --list                                                                 (or)
      memtool -l
 Test using the MEM test config file:=
      memtool --test --region=<region/'ALL'> [--testlist=<test0>,<test1>...]         (or)
      memtool -t -G <region/ALL> [-T <test0>,<test1>,...]
 Read the specified physical address:=
      memtool --read --addr=<address> --count=<bytes> [--width=<8/16/32>]            (or)
      memtool -r -a <address> -C <bytes> [-W <#8,16,32>]
 Write at the specified physical address:=
      memtool --write --addr=<address> --val=<data0>,<data1>, ... ,<dataN> [--
width=<8/16/32>] (or)
      memtool -w -a <address> -V <data0>,<data1>...,<dataN> [-W <8/16/32>]
 Execute repeatedly command by count:=
      memtool --iteration=max/<count> [option1] [option2]...                         (or)
      memtool -I max/<count> [option1] [option2]...

Usage:=
  -h, --h            Show the help text
  -t, --test         Test using the pre-programmed configuration or use supplied config
  -i, --info         Configuration information
  -l, --list         List the understood TLV codes and names
  -G, --region       Region
  -T, --testlist     List of tests
  -I, --iteration=   Iteration command execution
  -C, --count=       Count
  -a, --addr=        Address
  -r, --read         Read operation
```

```
  -w, --write        Write operation
  -V, --val=         Value to be set
  -W, --width        Width {8,16}
Available Tests are:
    ALL_TESTS, ADDRESS_READ, ADDRESS_WRITE, ADDRESS_WALKING1, ADDRESS_WALKING0, DATA_READ,
    DATA_WRITE, DATA_WALKING1, DATA_WALKING0, DATA_SLIDING1, DATA_SLIDING0, DATA_PATTERN,
    DATA_CACHE
e.g. ADDRESS_WALKING1,DATA_WALKING1
```

The `memtool` uses long options for the parameters that requires two hyphens in front of the options. Options are required, optional, or none. You use a parameter, it is specified as such and must include an equal sign; if an option is optional, it is enclosed with square brackets. However, do not type the brackets at the CLI. For example, the `-region` and `-testlist` options are optional and you must enter them as `-region=0` and `-testlist=0`.

- `List`—Lists the memory regions SDI knows. The tool queries SDI for the regions and prints a list of the regions with a region number that you can use for the subsequent options requiring a region number.
- `Info`—Lists the SPD information for the specified regions. Specifying a region allows the tool to read SPD from different DIMM modules, each specified in its own region. The output lists the actual data read and completes some parsing of the parameters so you do not have to decode the values. Decoding is based on the SPD standard definition for DDR3 and DDR4 DIMM memory.
- `Test`—Runs tests that include: Address Read/Write, Address Walking 1/0, Data Read/Write, Data Walking 1/0, Data Sliding 1/0, and Data Patterns (that writes Hamming patterns that you can use to detect multiple bit errors and identify single bit errors). These tests run during the normal memory tests. In extended memory tests, the data cache memory test runs. This test is lengthy and causes multiple ejections of data from the cache and tests the caches.

  In Verbosity 0, only the pass/fail message prints for all the tests. In Verbosity 1, each test prints its own pass/fail and other information; for example, what failed in the test. Higher verbosities show where each pass of the test performs and has verbose output. All output, regardless of verbosity, is in the log. You can see every level of detail by referring to the log.
- `Read`—Reads physical memory locations. You can loop over address read cycles to look for data that is volatile or read physical devices on the memory bus (`localbus` for Power-PC processors). You can specify a region, address, and count of successive bytes to read.
- `Write`—Writes to a physical memory address to test write cycles and memory. Similar to the `Read` command, this command takes a region, address in that region, and a comma-separated list of values to write.

Output

list Output

```
root@dell-diag-os:~# memtool --list
==================================
Region ID: 0
Region Name: DDR3-0
Address: dynamically allocated, Chunk: 0x2800 KB
Largest Cache Size: 0, Cache Line Size : 0
Access: d Increment: 8 Ecc: Y Iterations: 1
Configuration device: SPD (/dev/i2c-0) at 0x50, Regs 0 to 255
Tests:
Address Read Test
Address Write Test
Address Walking 1's Test
Address Walking 0's Test
Data Read Test
Data Write Test
Data Walking 1's Test
Data Walking 0's Test
Data Sliding 1's Test
Data Sliding 0's Test
Data Pattern Tests
Data Cache Test
root@dell-diag-os:~#
```

info Output

```
root@dell-diag-os:~# memtool --info
==== SPD Data ====
 Density 8192 MB,  Rows: 16, Cols: 10
 Bus Width: 64 bits, ECC: yes
 Manufacturer: Unknown
 Part Number : AW48M7228BNK0M
[00000000]: 0x92 0x13 0x0b 0x08 0x05 0x22 0x00 0x09 0x0b 0x11 0x01 0x08 0x0a 0x00 0xfe 0x00
  || .....".........
[00000010]: 0x69 0x78 0x69 0x3c 0x69 0x11 0x18 0x81 0xf0 0x0a 0x3c 0x3c 0x01 0x40 0x83 0x05
  || ixi<i.....<<.@..
[00000020]: 0x80 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x88 0x00 0x00 0x00 0x00 0x00 0x00
 || ................
[00000030]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0f 0x11 0x5f 0x00
 || ..............._.
[00000040]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 || ................
[00000050]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 || ................
[00000060]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 || ................
[00000070]: 0x00 0x00 0x00 0x00 0x00 0x86 0xe3 0x05 0x16 0x04 0xb3 0xd1 0x0d 0x05 0xec 0x10
 || ................
[00000080]: 0x41 0x57 0x34 0x38 0x4d 0x37 0x32 0x32 0x38 0x42 0x4e 0x4b 0x30 0x4d 0x00 0x00
 || AW48M7228BNK0M..
[00000090]: 0x00 0x00 0x00 0x00 0x00 0x00 0x41 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 || ......A.........
[000000a0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 || ................
[000000b0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[000000c0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[000000d0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[000000e0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[000000f0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
 || ...............
root@dell-diag-os:~#
```

test Output

```
root@dell-diag-os:~# memtool --test
Testing Memory Regions:
Testing Memory Region 0:
Address Read Test ..................................... Passed
Address Write Test .................................... Passed
Address Walking 1's Test .............................. Passed
Address Walking 0's Test .............................. Passed
Data Read Test ........................................ Passed
Data Write Test ....................................... Passed
Data Walking 1's Test ................................. Passed
Data Walking 0's Test ................................. Passed
Data Sliding 1's Test ................................. Passed
Data Sliding 0's Test ................................. Passed
Data Pattern Test ..................................... Passed
Memory: Overall test results ------------------------- >>> Passed
root@dell-diag-os:~#
```

read Output

```
root@dell-diag-os:~# memtool --read --addr=200
[00000200]: 0x00  || .
```

write Output

```
root@dell-diag-os:~# memtool --write --addr=200 --val=0x50
```

Constraints

You cannot perform memory tests while other tests that allocate and use memory within the region are performing. However, you can perform the `Read` tests concurrently with other processes. You cannot run multiple memory tests at the same time as they may collide within the memory spaces.

Memory tests cannot test all the memory, and without cache flushes, memory tests may not get out of the caches. The SDI must ensure the memory accessed is accessing the physical memory. This check slows down the tests.

Data flow

The `memtool` is not part of the data path and does not participate in the data flow.

nvramtool

To read and write the NVRAM bits that the BIOS uses to control testing and the bits for the EDA tools, use the `nvramtool`.

The NVRAM is an area, usually in a battery backed-up device such as an RTC chip, that allows the writing of bits that do not change across reboots or power cycles. These bits are used to control how devices boot and how the tests are performed. The `nvramtool` controls both the BIOS and EDA for testing. The bits are not common across platforms and are defined in the configuration file. When using this tool, you must write the correct bits because the tool does not know the details of the registers it is writing. The `nvramtool` can display the bit-level detail in the NVRAM registers, depending on how you define it in the configuration file.

Tests

There are no tests of the NVRAM. This tool only controls the bits.

CLI option

```
DellEmc Diag - NVRAM Tool
version 1.5, x.xx.x.x-x
build, 2017/05/23,

Syntax:  nvramtool <option>
  Show this help:=
      nvramtool --h                                          (or)
      nvramtool -h
  Read all or specfic register NVRAM values:=
      nvramtool --read [--reg=<register>]                    (or)
      nvramtool -r [-R <register>]
  Write NVRAM value:=
      nvramtool --write [--reg=<register> --val=<value>]     (or)
      nvramtool -w [-R <register> -V <value>]
  Execute repeatedly command by count:=
      nvramtool --iteration=max/<count> [option1] [option2]...(or)
      nvramtool -I max/<count> [option1] [option2]...

Usage:
      -h, --h           Show the help text
      -r, --read        Read operation
      -w, --write       Write operation
      -I, --iteration=  Iteration command execution
      -R, --reg=        Register
      -V, --val=        Value to be set
```

Output

read output

```
root@dell-diag-os:~# nvramtool --read
NVRAM Values:
0x00 0x9f 0x00 0xe6 0x03 0x03 0x00 0xea
Test Status Fail Bits : offset 0x50 = 0x0
    7 NVRAM test = 0
    6 SSD test = 0
    5 COLD/SMF Reg check = 0
    4 PCI test = 0
    3 Upper DRAM test = 0
    2 Lower DRAM test = 0
    1 ECC test = 0
    0 SPD test = 0
Test Status Pass Bits : offset 0x51 = 0x9f
    7 NVRAM test = 1
    6 SSD test = 0
    5 CPLD/SMF Reg check = 0
    4 PCI test = 1
    3 Upper DRAM test = 1
    2 Lower DRAM test = 1
    1 ECC test = 1
    0 SPD test = 1
RMT Control : offset 0x52 = 0x0
 7: 4 Undefined = 0
    3 RMT Test Enable = 0
 2: 0 RMT Test Reboot Count = 0
Status ID Byte : offset 0x53 = 0xe6
POST Control Bits : offset 0x54 = 0x3
    7 Force Cold Boot = 0
    6 POST Extended Upper DRAM test = 0
    5 POST Extended Lower DRAM test = 0
    4 POST Extended tests = 0
    3 Reserved = 0
    2 POST Verbose Mode = 0
    1 POST Stop on Error = 1
    0 POST Enable = 1
EDA Control Bits : offset 0x55 = 0x3
 5: 4 EDA Verbose Level = 0
    3 EDA Extended Tests = 0
    2 EDA Verbose Mode = 0
    1 EDA Stop on Error = 1
    0 EDA Enable = 1
EDA Extra Bits : offset 0x56 = 0x0
Control ID Byte : offset 0x57 = 0xea
root@dell-diag-os:~#
```

write output

```
./nvramtool --write --reg=0x54 --val=0x1
```

pcitool

To scan and access devices on the PCI bus, use the `pcitool`. The `pcitool` checks for missing devices and that the present devices are the proper type.

The `pcitool` scans the PCI bus for present devices and displays them and the BAR information it decodes. The tool does not handle endianess.

The `pcitool` reads the configuration file and then iterates across all devices in the configuration file. It checks the vendor/product ID to see that the correct device is at the correct address. The tool does not compare all the configuration space. The tool reads all 256 bytes of the configuration file.

Tests

The `pcitool` reads from the configuration file the devices it expects to find and reports any devices that it cannot find or if the device is not correct. The tool supports second-source parts; therefore, they are not flagged as false errors. If a mismatch occurs, the device lists with the expected value and the read value. Populate the configuration file with `-u` numbers so the device can quickly identify the failing device.

CLI options

```
DellEmc Diag - PCI Tool
version 1.5, x.xx.x.x-x
build, 2017/05/23,

Usage:
 To scan all PCI drivers and optionally show all config data :=
     pcitool --
scan[=all]                                                                    (or)
     pcitool -S[=all]
 To test using default PCI config-file :=
     pcitool --
test                                                                          (or)
     pcitool -t
 Show confic data for specific bus:dev.func:=
     pcitool --show {--bus=<bus># --dev=<dev># --
func=<func>#}                                        (or)
     pcitool -x {-B <bus># -D <dev># -F <func>#}
 Read 8-bit config register for bus:dev.func:=
     pcitool --read {--bus=<bus># --dev=<dev># --func=<func># --offset=<offset> --
count=<count>} (or)
     pcitool -r {-B <bus># -D <dev># -F <func># -O <offset> -C <count>}
 Write 8-bit config register for bus:dev.func:=
     pcitool --write {--bus=<bus># --dev=<dev># --func=<func># --offset=<offset> --
val=<value>}  (or)
     pcitool -w {-B <bus># -D <dev># -F <func># -O <offset> -V <value>}
 Execute repeatedly command by count:=
     pcitool --iteration=max/<count> [option1]
[option2]...                                          (or)
     pcitool -I max/<count> [option1] [option2]...

Syntax: pcitool <option>
   -h, --h             Show the help text
   -S, --scan          Scan operation
   -t, --test          Test using the pre-programmed configuration or use supplied config
   -x, --show          Show operation
   -r, --read          Read operation
   -w, --write         Write operation
   -I, --iteration=    Iteration command execution
   -B, --bus=          To specify the i2c bus   e.g.: /dev/i2c-<bus number>
   -D, --dev=          Device
   -F, --func=         Func
   -O, --offset=       Set the Offset
   -C, --count=        Count
   -V, --val=          Value to be set
```

Output

scan output

```
root@dell-diag-os:~# pcitool --scan
Acquiring PCI device name database
Device#01:  bus:dev.fn 00:00.0 - ID=0x1f0c8086, Intel Atom Processor SoC Transaction Router
Device#02:  bus:dev.fn 00:01.0 - ID=0x1f108086, Intel Atom Processor PCIe Root Port 1
Device#03:  bus:dev.fn 00:02.0 - ID=0x1f118086, Intel Atom Processor PCIe Root Port 2
Device#04:  bus:dev.fn 00:03.0 - ID=0x1f128086, Intel Atom Processor PCIe Root Port 3
```

```
Device#05:  bus:dev.fn 00:04.0 - ID=0x1f138086, Intel Atom Processor PCIe Root Port 4
Device#06:  bus:dev.fn 00:0e.0 - ID=0x1f148086, Intel Atom Processor C2000 RAS
Device#07:  bus:dev.fn 00:0f.0 - ID=0x1f168086, Intel Atom Processor C2000 RCEC
Device#08:  bus:dev.fn 00:13.0 - ID=0x1f158086, Intel Atom processor C2000 SMBus 2.0
Device#09:  bus:dev.fn 00:14.0 - ID=0x1f418086, Intel Ethernet Connection I354
Device#10:  bus:dev.fn 00:14.1 - ID=0x1f418086, Intel Ethernet Connection I354
Device#11:  bus:dev.fn 00:14.2 - ID=0x1f418086, Intel Ethernet Connection I354
Device#12:  bus:dev.fn 00:16.0 - ID=0x1f2c8086, Intel USB Enhanced Host Controller
Device#13:  bus:dev.fn 00:17.0 - ID=0x1f228086, Intel AHCI SATA2 Controller
Device#14:  bus:dev.fn 00:18.0 - ID=0x1f328086, Intel AHCI SATA3 Controller
Device#15:  bus:dev.fn 00:1f.0 - ID=0x1f388086, Intel ISA bridge
Device#16:  bus:dev.fn 00:1f.3 - ID=0x1f3c8086, Intel PCU SMBus
Device#17:  bus:dev.fn 01:00.0 - ID=0x837514e4, Broadcom Network Processor BCM88375
Device#18:  bus:dev.fn 01:00.1 - ID=0x837514e4, Broadcom Network Processor BCM88375
root@dell-diag-os:~#
```

test output

```
root@dell-diag-os:~# pcitool --test
Testing PCI devices:
+ Checking PCI 00:00.0, ID=1f0c8086 ................. Passed
+ Checking PCI 00:01.0, ID=1f108086 ................. Passed
+ Checking PCI 00:02.0, ID=1f118086 ................. Passed
+ Checking PCI 00:03.0, ID=1f128086 ................. Passed
+ Checking PCI 00:0e.0, ID=1f148086 ................. Passed
+ Checking PCI 00:0f.0, ID=1f168086 ................. Passed
+ Checking PCI 00:13.0, ID=1f158086 ................. Passed
+ Checking PCI 00:14.0, ID=1f418086 ................. Passed
+ Checking PCI 00:14.1, ID=1f418086 ................. Passed
+ Checking PCI 00:14.2, ID=1f418086 ................. Passed
+ Checking PCI 00:16.0, ID=1f2c8086 ................. Passed
+ Checking PCI 00:17.0, ID=1f228086 ................. Passed
+ Checking PCI 00:18.0, ID=1f328086 ................. Passed
+ Checking PCI 00:1f.0, ID=1f388086 ................. Passed
+ Checking PCI 00:1f.3, ID=1f3c8086 ................. Passed
+ Checking PCI 01:00.0, ID=837514e4 ................. Passed
+ Checking PCI 01:00.1, ID=837514e4 ................. Passed
PCI devices: Overall test results --------------- >>> Passed
root@dell-diag-os:~#
```

show output

```
root@dell-diag-os:/etc/dn/diag# pcitool --show --bus=0 --dev=4 --func=0bus
bus:dev.fn 00:04.3
[00000000]: 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[00000010]: 0x40 0x0e 0x40 0x00 0x00 0x00 0x00 0x00 0xe5 0xe2 0xdd 0x5b 0x47 0x7f 0x00 0x00
  || @.@.........[G...
[00000020]: 0xff 0xff 0xff 0xff 0x00 0x00 0x00 0x00 0x0c 0x00 0xad 0xfb 0x00 0x00 0x00 0x00
  || ................
[00000030]: 0xf0 0x30 0x5f 0x02 0x00 0x00 0x00 0x00 0x10 0x30 0x5f 0x02 0x00 0x00 0x00 0x00
  || .0_......0_.....
[00000040]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x40 0x0e 0x40 0x00 0x00 0x00 0x00 0x00
  || ........@.@.....
[00000050]: 0x80 0xa0 0xa9 0x91 0xff 0x7f 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[00000060]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x35 0x37 0x86 0x5b 0x47 0x7f 0x00 0x00
  || ........57.[G...
[00000070]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x20 0x9f 0xa9 0x91 0xff 0x7f 0x00 0x00
  || ........ .......
[00000080]: 0x40 0x0e 0x40 0x00 0x00 0x00 0x00 0x00 0xe4 0x1b 0x40 0x00 0x00 0x00 0x00 0x00
  || @.@.......@.....
[00000090]: 0x04 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x5c 0x9f 0xa9 0x91 0xff 0x7f 0x00 0x00
  || ........\.......
[000000a0]: 0xda 0x4e 0x40 0x00 0x00 0x00 0x00 0x00 0x20 0xbe 0xa9 0x91 0x00 0x7f 0x00 0x00
  || .N@..... .......
[000000b0]: 0xa0 0x9f 0xa9 0x91 0x00 0x00 0x00 0x00 0x10 0x30 0x5f 0x02 0x00 0x00 0x00 0x00
```

```
  || .........0_.....
[000000c0]: 0x30 0x34 0x2e 0x30 0x00 0x74 0x65 0x73 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || 04.0.tes........
[000000d0]: 0x2f 0x70 0x72 0x6f 0x63 0x2f 0x62 0x75 0x73 0x2f 0x70 0x63 0x69 0x2f 0x30 0x30
  || /proc/bus/pci/00
[000000e0]: 0x2f 0x30 0x34 0x2e 0x30 0x00 0x00 0x00 0x80 0xa0 0xa9 0x91 0xff 0x7f 0x00 0x00
  || /04.0...........
[000000f0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
  Base Address 0: Memory at 0x00400e40.
  Base Address 1: Memory at 0x00000000.
  Base Address 2: I/O at 0x5bdde2e0.
  Base Address 3: I/O at 0x00007f40.
  Base Address 4: I/O at 0xfffffff0.
  Base Address 5: Memory at 0x00000000.
  CardBus CIS pointer 0xfbad000c (BAR 3), address 7f47.
root@dell-diag-os:/etc/dn/diag# pcitool --show --bus=0 --dev=4 --func=0
bus:dev.fn 00:04.0
[00000000]: 0x86 0x80 0x13 0x1f 0x07 0x04 0x10 0x00 0x02 0x00 0x04 0x06 0x10 0x00 0x01 0x00
  || ................
[00000010]: 0x04 0x00 0xf6 0xdf 0x00 0x00 0x00 0x00 0x00 0x04 0x04 0x00 0xf0 0x00 0x00 0x20
  || ................
[00000020]: 0xf0 0xff 0x00 0x00 0xf1 0xff 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[00000030]: 0x00 0x00 0x00 0x00 0x40 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x07 0x01 0x10 0x00
  || ....@...........
[00000040]: 0x10 0x80 0x42 0x01 0x21 0x80 0x00 0x00 0x0f 0x20 0x00 0x00 0x42 0x48 0x79 0x04
  || ..B.!.... ..BHy.
[00000050]: 0x40 0x00 0x01 0x10 0x00 0xfd 0x18 0x00 0xc0 0x03 0x00 0x00 0x08 0x00 0x00 0x00
  || @................
[00000060]: 0x00 0x00 0x00 0x00 0xb7 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x06 0x00 0x00 0x00
  || ................
[00000070]: 0x02 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[00000080]: 0x01 0x88 0x03 0xc8 0x00 0x00 0x00 0x00 0x0d 0x90 0x00 0x00 0x86 0x80 0x86 0x80
  || ................
[00000090]: 0x05 0x00 0x01 0x01 0x0c 0xf0 0xe0 0xfe 0xa1 0x41 0x00 0x00 0x00 0x00 0x00 0x00
  || .........A......
[000000a0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[000000b0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
[000000c0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00
  || ................
[000000d0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x80 0x00 0x00 0x00 0x00
  || ................
[000000e0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x00
  || ................
[000000f0]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ................
  Base Address 0: Memory at 0xdff60000.
  Base Address 1: Memory at 0x00000000.
  Base Address 2: Memory at 0x00040400.
  Base Address 3: Memory at 0x200000f0.
  Base Address 4: Memory at 0x0000fff0.
  Base Address 5: I/O at 0x0001fff0.
  Address 0 at 0xdff60000, 64 bit
  Address 2 at 0x00040400, 32 bit
  Address 3 at 0x200000f0, 32 bit
  Address 4 at 0x0000fff0, 32 bit
  Extended capabilities, first structure at offset 0x40.
  Extended PCI capability type 16 at 0x40, next 128.
  Extended PCI capability type 1 at 0x80, next 136.
   Power management entry ver. 3: Capabilities c803, Ctrl 0000, Event 0000.
   Power state D0.
  Extended PCI capability type 13 at 0x88, next 144.
  Extended PCI capability type 5 at 0x90, next 0.
root@dell-diag-os:/etc/dn/diag#
```

phytool

The phytool allows setting the management phy for management port for speed, duplex auto negotiation, and Loopback; as well as reading the MAC and MAC EEPROM in the phy.

Tests

CLI options

```
DellEmc Diag - PHY Tool
version 1.1, x.xx.x.x-x
build, 2017/05/23,

Syntax: phytool <option>
  Show the help-text:=
       phytool --h                                                    (or)
       phytool -h
  Read the mac address of the interface:=
       phytool --read-mac                                             (or)
       phytool -R
  Write the value to the specified offset:=
       phytool --write --offset=<offset> --val=<val>                  (or)
       phytool -w -o <offset> -V <val>
  Dump the eeprom contents:=
       phytool --eeprom-dump                                          (or)
       phytool -x
  Dump the register contents:=
       phytool --reg-dump                                             (or)
       phytool -d
  Phy loopback test:=
       phytool --lb-test[=no of packets]                              (or)
       phytool -l[=no of packets]
  Execute repeatedly command by count:=
       phytool --iteration=max/<count> [option1] [option2]...         (or)
       phytool -I max/<count> [option1] [option2]...
  Set the interface with parameters:=
       phytool --set-intf --speed=<speed> --duplex=<mode> --autoneg (or)
       phytool -s -S <speed> -D <mode> -A
  Show the interface settings:=
       phytool --show-intf                                            (or)
       phytool -a
  Usage:=
       -h, --h              Show the help text
       -I, --iteration=     Iteration command execution
       -R, --read-mac       Read the MAC of the interface
       -w, --write          Write operation
       -o, --offset         Set the Offset
       -V, --val            Value to be set
       -x, --eeprom-dump    Dump the eeprom contents
       -d, --reg-dump       Dump the register contents
       -l, --lb-test=       Phy loopback test
       -s, --set-intf       Set the interface with parameters
       -S, --speed=         Speed
       -D, --duplex=        Duplex mode
       -A, --autoneg=       Auto-negotiation
       -a, --show-intf      Show the interface settings
```

Output

```
root@dellemc-diag-os:/etc/dn/diag# phytool --read-mac
34:17:eb:07:7c:00

root@dellemc-diag-os:/etc/dn/diag# phytool --eeprom-dump
Offset      Values
```

```
------      ------
0x0000:     34 17 eb 07 7c 00 00 08 ff ff 05 10 ff ff ff ff
0x0010:     18 00 00 00 2f 40 41 1f 86 80 41 1f 86 80 80 ba
0x0020:     ff ff ff ff 80 5c 47 00 00 00 40 00 00 4c ab 03
0x0030:     00 00 00 70 0e 1a 26 44 a3 07 42 1f 01 02 02 06
0x0040:     0c 00 47 21 00 00 ff ff ac 44 f6 00 44 1f 08 09
0x0050:     40 04 3c 00 00 00 04 14 00 00 00 00 00 10 ff ff
0x0060:     00 01 00 40 32 13 13 40 00 01 00 40 ff ff b0 03
0x0070:     00 01 00 40 00 01 00 40 d9 09 bc 03 ff ff b5 7e
0x0080:     ff ff ff ff a5 0b 00 80 ff ff ff ff ff ff ff ff
.....

root@dellemc-diag-os:/etc/dn/diag# phytool --reg-dump
0x00000: CTRL (Device control register)          0x08100241
       Invert Loss-Of-Signal:                    no
       Receive flow control:                     enabled
       Transmit flow control:                    disabled
       VLAN mode:                                disabled
       Set link up:                              1
       D3COLD WakeUp capability advertisement:   enabled
       Auto speed detect:                        disabled
       Speed select:                             1000Mb/s
       Force speed:                              no
       Force duplex:                             no
0x00008: STATUS (Device status register)         0x00282383
       Duplex:                                   full
       Link up:                                  link config
       Transmission:                             on
       DMA clock gating:                         disabled
       TBI mode:                                 disabled
       Link speed:                               1000Mb/s
       Bus type:                                 PCI Express
...

root@dellemc-diag-os:/etc/dn/diag# phytool --lb-test=100
TEST PASSED

NOTE: The loopback test and set-intf will terminate the ethernet driver.  You need to reboot
to restart the driver cleanly.
[1]+  Terminated              setsid /bin/kni -c 0x3 -n 2 -- -p 1 --config="(0,0,1)" >> /dev/
null

root@dellemc-diag-os:~# phytool --set-intf --speed=1000
[2]+  Done                    dhclient -q eth0
root@dellemc-diag-os:~# .......................done
Port 0 Link Up - speed 1000 Mbps - full-duplex

root@dellemc-diag-os:~# root@dellemc-diag-os:~# phytool --show-intf
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: Symmetric
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 3
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: off (auto)
        Supports Wake-on: pumbg
        Wake-on: g
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

pltool

To test functionality of the CPLD and FPGA devices on the boards during startup, use the `pltool`.

The `pltool` also checks for the correct firmware loads. The tool uses the CLI to list the devices and their registers, and allows you to read and write registers in the device. The read functionality prints the details to the bit level and also any bit groupings and their meanings. The tool uses the SDI interface to get a list of devices and registers in the system, and then uses SDI to access the devices.

Tests

The `pltool` tests specified registers and values SDI identifies in the testable bits of the register. The tool reads the register using SDI interfaces and compares the testable bits from those bits the SDI database provides. If a mismatch occurs, an error displays.

CLI options

```
Syntax: pltool <option>
  Show this help text:=
       pltool --h                                                                    (or)
       pltool -h
  Test (RW) the scratchpad registers:=
       pltool --test                                                                 (or)
       pltool -t
  List devices and registers:=
       pltool --list [--lstype=<devicetype>]                                         (or)
       pltool -l [-T <devicetype>]
  List device names with address:=
       pltool --listdevicenames  [--devname=<devicename>]                            (or)
       pltool -L [-n <devicename>]
  Execute repeatedly command by count:=
       pltool --iteration=max/<count> [option1] [option2]...                         (or)
       pltool -I max/<count> [option1] [option2]...
  Read the specified register of the device:=
       pltool --read --devname=<devicename> --dev=<deviceaddr> --reg=<register>      (or)
       pltool -r -n <devicename> -D <deviceaddr> -R <register>
  Write at the specified register of the device:=
       pltool --write --devname=<devicename> --dev=<deviceaddr> --reg=<register> --
val=<value> (or)
       pltool -w -n <devicename> -D <deviceaddr> -R <register> -V <value>
  Dump all of the registers in a device or all devices and thei rcurrent values:=
       pltool --dump [--devname=<devicename>] [--dev=<deviceaddr>] (or)
       pltool -d [-n <devicename>] [-D <deviceaddr>]

Usage:=
  -h, --h               Show the help text
  -t, --test            Test using the pre-programmed configuration or use supplied config
  -l, --list            List the understood TLV codes and names
  -T, --lstype          Device type
  -L, --listdevicenames List Device name
  -r, --read            Read operation
  -w, --write           Write operation
  -I, --iteration=      Iteration command execution
  -n, --devname=        Device name
  -D, --dev=            Device
                        (should be assigned 0 for lpc access)
  -R, --reg=            Register
  -V, --val=            Value to be set
  -d, --dump            Dump the values in the registers of a device
```

Output

list output

```
root@dell-diag-os:~# pltool --list
CPLD1 0 cpld lpc 0 (U5)
   0x100 CPLD_VERSION bits:8 RO val:0 mask:0xff test:0 ver:0x0
      7:4 MAJOR_VER RO 0
      3:0 MINOR_VER RO 0
   0x101 BOARD_TYPE bits:8 RO val:0xff mask:0xff test:0 ver:0x0
      7:0 BOARD_TYPE RO 0x1
          3 <platform> Board
   0x102 SW_SCRATCH bits:8 RW val:0xde mask:0xff test:1 ver:0x0
      7:0 SW_SCRATCH RW 0xde
   0x103 CPLD_ID bits:8 RO val:0xff mask:0xff test:0 ver:0x0
      7:0 CPLD_ID RO 0x1
   0x10f BOARD_REV bits:8 RO val:0xff mask:0xff test:0 ver:0x0
      7:0 BOARD_REV RO 0
   0x110 CPLD_SEP_RST0 bits:8 RO val:0xff mask:0xff test:0 ver:0x0
         7 Reset Extender CPLD 4 RW 0x1
          0 Reset
          1 Not Reset
         6 Reset Extender CPLD 3 RW 0x1
          0 Reset
          1 Not Reset
         5 Reset Extender CPLD 2 RW 0x1
          0 Reset
          1 Not Reset
         4 PCA9548_RST7 RW 0x1
          0 Reset
          1 Not Reset
         3 PCA9548_RST6 RW 0x1
          0 Reset
          1 Not Reset
         2 PCA9548_RST5 RW 0x1
          0 Reset
          1 Not Reset
         1 PCA9548_RST4 RW 0x1
          0 Reset
          1 Not Reset
```

listdevicenames output

Based from the output of `--devicenames`, you can decide if you need to use the `--devname= option` in the read or write functions. You can access `CPLD1` being at `deviceaddress 0`, using the register value for the register you want, such as:

```
root@dell-diag-os:~# pltool -listdevicenames
0x0 : CPLD1
0x3e : CPLD2
0x3e : CPLD3
0x3e : CPLD4
0x0 : SMF_FPGA
```

read output

```
root@dell-diag-os:~# pltool --read --devname=CPLD4 --dev=0x3e --reg=0x2
SW_SCRATCH : offset 0x02 = 0xde
 7: 0 SW_SCRATCH = de
root@dell-diag-os:~#
```

write output

```
root@dell-diag-os:~# pltool --write  --devname=CPLD4 --dev=0x3e --reg=0x2 --val=0xff
```

test output

```
root@dell-diag-os:~# pltool --test
Testing Programmable Devices:
PL Tool test:
 CPLD1 ..................... Passed
 CPLD2: SW_SCRATCH..................... Passed
 CPLD3: SW_SCRATCH..................... Passed
 CPLD4: SW_SCRATCH..................... Passed
 SMF_FPGA ............................. Passed
PL Tool: Overall test results ---- >>> Passed
```

psutool

The `psutool` determines which PSUs are in the system, checks the Power Good setting, and reads the field replaceable unit (FRU) information. It does not look at the PSU fans and airflow direction of the fans.

Tests

The `psutool` looks for the presence of the PSU and if the PSU is present, it checks the Power Good setting in the CPLD. It does not read directly from the PSU but reads the CPLD information instead. If the PSU is present and it does not receive a Power Good signal, it does not know if the power plug is not installed or if the PSU is not operating correctly, so it displays a failure.

CLI options

```
DellEmc Diag - Power Supply Tool
version 1.4, x.xx.x.x-x
build, 2017/05/23,

Syntax: psutool <option>
 Show the Help-text:=
        psutool --h                                                              (or)
        psutool -h
 Test using the default config file:=
        psutool --test [--supply=<power_supply>]                                 (or)
        psutool -t [-S <power_supply>]
 Read the register on the Power Supply:=
        psutool --read --supply=<power_supply> --reg=<register>                  (or)
        psutool -r -S <power_supply> -r <register>
 Write the value into the Power Supply Register:=
        psutool --write --supply=<power_supply> --reg=<register> --val=<value> (or)
        psutool -w <power_supply> -R <register> -V <value>
 Verify PSU by reading SMF registers:=
        psutool --lpc                                                            (or)
        psutool -q
 Execute repeatedly command by count:=
        psutool --iteration=max/<count> [option1] [option2]...                   (or)
        psutool -I max/<count> [option1] [option2]...

Usage:=
   -h, --h          Show the help text
   -t, --test       Test using the pre-programmed configuration or use supplied config
   -S, --supply=    Power supply
   -r, --read       Read operation
   -w, --write      Write operation
   -R, --register=  Register
   -V, --value=     Value to be set
   -I, --iteration= Iteration command execution
```

```
   -q, --lpc        Verify PSU by reading SMF registers.
                    This option must be used along with test flag
```

test option

```
root@dell-diag-os:~# psutool --test --lpc
Power Supply Test all
Getting details of Power Supply 1 using LPC interface
Power Supply 1 is Present
Power Supply 1 Input Type AC
Power Supply 1 Input Voltage(VIN) : 203.250000 V
Power Supply 1 Output Voltage(VOUT) : 12.210000 V
Power Supply 1 Input Current(IIN) : 0.610000 A
Power Supply 1 Output Current(IOUT) : 9.150000 A
Power Supply 1 Input Power(PIN) : 124.000000 W
Power Supply 1 Output Power(POUT) : 111.700000 W
Power Supply 1 Temperature : 30.000000 C
Power Supply 1 Fan Present
Power Supply 1 Fan Status is Normal
Power Supply 1 Fan Airflow Type is F2B
Power Supply 1 Fan Speed(RPM) : 9072
Getting details of Power Supply 2 using LPC interface
Power Supply 2 is Present
Power Supply 2 Input Type AC
Power Supply 2 Output Voltage Low
Power Supply 2 Input Voltage(VIN) : 0.000000 V
Power Supply 2 Output Voltage(VOUT) : 0.000000 V
Power Supply 2 Input Current(IIN) : 0.000000 A
Power Supply 2 Output Current(IOUT) : 0.000000 A
Power Supply 2 Input Power(PIN) : 0.000000 W
Power Supply 2 Output Power(POUT) : 0.000000 W
Power Supply 2 Temperature : 6553.100098 C
Power Supply 2 Fan Present
Power Supply 2 Fan Status is Normal
Power Supply 2 Fan Airflow Type is F2B
Power Supply 2 Fan Speed(RPM) : 9120
Power Supply Test ..................................... Passed
root@dell-diag-os:~#
```

rtctool

The rtctool allows setting and testing of the real time clock (RTC) in the system.

Tests

CLI options

```
DellEmc Diag - RTC Tool
version 1.1, x.xx.x.x-x
build, 2017/05/23,

Syntax: rtctool <option>
   Show the help-text:=
       rtctool --help                          (or)
       rtctool -h
   Read the current RTC:=
       rtctool --readrtc                       (or)
       rtctool -r
   Test RTC device with user interrupt:=
       rtctool --testuie                       (or)
       rtctool -u
   Test RTC device with alarm interrupt:=
       rtctool --testaie                       (or)
       rtctool -a
```

```
   Test RTC device with periodic interrupt:=
       rtctool --testpie                          (or)
       rtctool -p
   Test the RTC device:=
       rtctool --test                             (or)
       rtctool -t
   Set rtc to new time (input all params in same order):=
       rtctool --setrtc --year=<year>, --mon=<month> --day=<day> --hour=<hour> --min=<min> --
sec=<sec> --tz=<offset>
                        (or)
       rtctool -s -y <year> -m <month> -D <day> -H <hour> -M <min> -S <sec> -Z <offset>
   Execute repeatedly command by count:=
       rtctool --iteration=max/<count> [option1] [option2]... (or)
       rtctool -I max/<count> [option1] [option2]...

Usage:=
       -h, --help         Show the help text
       -r, --readrtc      Read operation
       -s, --setrtc       Set operation
       -u, --testuie      Test RTC device with user interrupt
       -a, --testaie      Test RTC device with alarm interrupt
       -p, --testpie      Test RTC device with periodic interrupt
       -I, --iteration=   Iteration command execution
       -y, --year=        Year
       -m, --month=       Month
       -D, --day=         Day
       -H, --hour=        Hour
       -M, --min=         Minute
       -S, --sec=         Second
       -Z, --offset=      +12.0 to -12.0 timezone offset
```

storagetool

The `storagetool` tests mounted storage media.

The tool searches for any device in `/dev/hd*`, `sda`, `sdb`, or `sdc` and tests using them. The tests are file-copy tests to the device in the mounted file system. The files are written, compared and removed, leaving the file system as it was before the test. You can run more tests using the `bonnie++ tool` and the tool reads SMART data from the device using the `smart` option.

Tests

The standard test creates a `directory` on the file system, opens a file for write, copies the file, compares the files, and reports errors. The test repeats 10 times. After the test completes successfully, `storagetool` removes all the test files.

CLI options

```
DellEmc Diag - Storage Tool
version 1.1, x.xx.x.x-x
build, 2017/05/23,

Syntax: storagetool <option>
 Show the help-text:=
         storagetool --h                        (or)
         storagetool -h
 Mount usb device when inserted (mandatory):=
         storagetool --mountusb                 (or)
         storagetool -m
 Unmount usb device before removed (mandatory):=
         storagetool --unmountusb               (or)
         storagetool -u
 List devices:=
         storagetool --list                     (or)
         storagetool -l
 Test devices(empty for all):=
         storagetool --test [--dev=<device>]    (or)
         storagetool -t [-D <device>
```

```
 Get the smart status for a device
         storagetool --smart --dev=<device>     (or)
         storagetool -S -D <device>
 Execute repeatedly command by count:=
         storagetool --iteration=max/<count> [option1] [option2]...(or)
         storagetool -I max/<count> [option1] [option2]...
 Run the bonnie tools on the filesystems:=
         storagetool --bonnie                   (or)
         storagetool -B

Usage:
   -h, --h           Show the help text
   -m, --mountusb    Mount usb device when inserted (mandatory)
   -u, --unmountusb  Unmount usb device when inserted (mandatory)
   -l, --list        List all storage devices
   -S, --smart       Smart Status
   -D, --dev=        Device
   -T, --test        Test using the pre-programmed configuration or use supplied config
   -I, --iteration=  Iteration command execution
   -B, --bonnie      Run the bonnie tools on the filesystems
```

Output

list output

```
root@dell-diag-os:~# storagetool --list
Mounted Filesystem Devices:
/dev/sda3 / ext4
root@dell-diag-os:~#
```

test output

```
root@dell-diag-os:~# storagetool --test --dev=/dev/sda3
Testing Storage Devices ....................... Passed
root@dell-diag-os:~#
```

smart output

```
root@dell-diag-os:~# storagetool --smart --dev=/dev/sda3
smartctl 6.2 2013-07-26 r3841 [x86_64-linux-3.15.10] (local build)
Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Device Model:     InnoDisk Corp. - mSATA 3IE
Serial Number:    20160119AA144700000F
Firmware Version: S141002c
User Capacity:    32,017,047,552 bytes [32.0 GB]
Sector Size:      512 bytes logical/physical
Rotation Rate:    Solid State Device
Device is:        Not in smartctl database [for details use: -P showall]
ATA Version is:   ACS-2 (minor revision not indicated)
SATA Version is:  SATA 3.0, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is:    Mon Jan  1 20:45:44 2001 UTC
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF ENABLE/DISABLE COMMANDS SECTION ===
SMART Enabled.

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
```

```
General SMART Values:
Offline data collection status:  (0x00) Offline data collection activity
                                        was never started.
                                        Auto Offline Data Collection: Disabled.
Total time to complete Offline
data collection:                 (   32) seconds.
Offline data collection
capabilities:                    (0x00)         Offline data collection not supported.
SMART capabilities:              (0x0003) Saves SMART data before entering
                                        power-saving mode.
                                        Supports SMART auto save timer.
Error logging capability:        (0x00) Error logging NOT supported.
                                        General Purpose Logging supported.
SCT capabilities:                (0x0039) SCT Status supported.
                                        SCT Error Recovery Control supported.
                                        SCT Feature Control supported.
                                        SCT Data Table supported.

SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED
RAW_VALUE
  1 Raw_Read_Error_Rate     0x0000   000   000   000    Old_age   Offline      -       0
  2 Throughput_Performance  0x0000   000   000   000    Old_age   Offline      -       0
  3 Spin_Up_Time            0x0000   000   000   000    Old_age   Offline      -       0
  5 Reallocated_Sector_Ct   0x0002   100   100   000    Old_age   Always       -       0
  7 Unknown_SSD_Attribute   0x0000   000   000   000    Old_age   Offline      -       0
  8 Unknown_SSD_Attribute   0x0000   000   000   000    Old_age   Offline      -       0
  9 Power_On_Hours          0x0002   100   100   000    Old_age   Always       -       3289
 10 Unknown_SSD_Attribute   0x0000   000   000   000    Old_age   Offline      -       0
 12 Power_Cycle_Count       0x0002   100   100   000    Old_age   Always       -       205
168 Unknown_Attribute       0x0000   000   000   000    Old_age   Offline      -       0
169 Unknown_Attribute       0x0000   000   000   000    Old_age   Offline      -       0
175 Program_Fail_Count_Chip 0x0000   000   000   000    Old_age   Offline      -       0
192 Power-Off_Retract_Count 0x0000   000   000   000    Old_age   Offline      -       0
  1 Raw_Read_Error_Rate     0x0000   000   000   000    Old_age   Offline      -
2199023255552
197 Current_Pending_Sector  0x0000   000   000   000    Old_age   Offline      -       0
240 Unknown_SSD_Attribute   0x0000   000   000   000    Old_age   Offline      -       0
225 Unknown_SSD_Attribute   0x0000   000   000   000    Old_age   Offline      -       0
170 Unknown_Attribute       0x0003   100   100   ---    Pre-fail  Always       -       1966080
173 Unknown_Attribute       0x0002   100   100   ---    Old_age   Always       -       7602213
229 Unknown_Attribute       0x0002   100   100   ---    Old_age   Always       -
88470212370072
236 Unknown_Attribute       0x0002   100   100   ---    Old_age   Always       -       0
235 Unknown_Attribute       0x0002   100   000   ---    Old_age   Always       -       0
176 Erase_Fail_Count_Chip   0x0000   100   000   ---    Old_age   Offline      -       0

Read SMART Log Directory failed: scsi error aborted command

Read SMART Error Log failed: scsi error aborted command

Read SMART Self-test Log failed: scsi error aborted command

Selective Self-tests/Logging not supported

root@dell-diag-os:~#
```

bonnie output

```
root@dell-diag-os:~# storagetool --bonnie --dev=/dev/sda3
Using uid:0, gid:0.
Writing with putc()...done
Writing intelligently...done
Rewriting...done
Reading with getc()...done
Reading intelligently...done
start 'em...done...done...done...
Create files in sequential order...done.
Stat files in sequential order...done.
```

```
Delete files in sequential order...done.
Create files in random order...done.
Stat files in random order...done.
Delete files in random order...done.
Version  1.03       ------Sequential Output------ --Sequential Input- --Random-
                    -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine        Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP  /sec %CP
dell-diag-os   250M 27664  96 245045  62 +++++ +++ 31064 100 +++++ +++ +++++ +++
                    ------Sequential Create------ --------Random Create--------
                    -Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
              files  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP  /sec %CP
                 32 32494  97 +++++ +++ 31198  66 31739  92 +++++ +++ 26511  56
dell-diag-os,250M,27664,96,245045,62,+++++,+++,31064,100,+++++,+++,+++++,+++,32,32494,97,++++
+,+++,31198,66,31739,92,+++++,+++,26511,56
```

smartctl

To get a usage summary, use the `smartctl -h` command.

```
root@dell-diag-os:/opt/dell/diag/bin# smartctl -h
smartctl 6.2 2013-07-26 r3841 [x86_64-linux-3.15.10] (local build)
Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org

Usage: smartctl [options] device

============================================ SHOW INFORMATION OPTIONS =====

  -h, --help, --usage
         Display this help and exit

  -V, --version, --copyright, --license
         Print license, copyright, and version information and exit

  -i, --info
         Show identity information for device

  --identify[=[w][nvb]]
         Show words and bits from IDENTIFY DEVICE data                (ATA)

  -g NAME, --get=NAME
        Get device setting: all, aam, apm, lookahead, security, wcache, rcache, wcreorder

  -a, --all
         Show all SMART information for device

  -x, --xall
         Show all information for device

  --scan
         Scan for devices

  --scan-open
         Scan for devices and try to open each device

================================== SMARTCTL RUN-TIME BEHAVIOR OPTIONS =====

  -q TYPE, --quietmode=TYPE                                           (ATA)
         Set smartctl quiet mode to one of: errorsonly, silent, noserial

  -d TYPE, --device=TYPE
         Specify device type to one of: ata, scsi, sat[,auto][,N][+TYPE], usbcypress[,X],
usbjmicron[,p][,x][,N], usbsunplus, marvell, areca,N/E, 3ware,N, hpt,L/M/N, megaraid,N,
cciss,N, auto, test

  -T TYPE, --tolerance=TYPE                                           (ATA)
         Tolerance: normal, conservative, permissive, verypermissive

  -b TYPE, --badsum=TYPE                                              (ATA)
         Set action on bad checksum to one of: warn, exit, ignore

  -r TYPE, --report=TYPE
```

```
         Report transactions (see man page)

  -n MODE, --nocheck=MODE                                               (ATA)
         No check if: never, sleep, standby, idle (see man page)

============================== DEVICE FEATURE ENABLE/DISABLE COMMANDS =====

  -s VALUE, --smart=VALUE
        Enable/disable SMART on device (on/off)

  -o VALUE, --offlineauto=VALUE                                         (ATA)
        Enable/disable automatic offline testing on device (on/off)

  -S VALUE, --saveauto=VALUE                                            (ATA)
        Enable/disable Attribute autosave on device (on/off)

  -s NAME[,VALUE], --set=NAME[,VALUE]
        Enable/disable/change device setting: aam,[N|off], apm,[N|off],
        lookahead,[on|off], security-freeze, standby,[N|off|now],
        wcache,[on|off], rcache,[on|off], wcreorder,[on|off]

=================================== READ AND DISPLAY DATA OPTIONS =====

  -H, --health
        Show device SMART health status

  -c, --capabilities                                                    (ATA)
        Show device SMART capabilities

  -A, --attributes
        Show device SMART vendor-specific Attributes and values

  -f FORMAT, --format=FORMAT                                            (ATA)
        Set output format for attributes: old, brief, hex[,id|val]

  -l TYPE, --log=TYPE
        Show device log. TYPE: error, selftest, selective, directory[,g|s],
                               xerror[,N][,error], xselftest[,N][,selftest],
                               background, sasphy[,reset], sataphy[,reset],
                               scttemp[sts,hist], scttempint,N[,p],
                               scterc[,N,M], devstat[,N], ssd,
                               gplog,N[,RANGE], smartlog,N[,RANGE]

  -v N,OPTION , --vendorattribute=N,OPTION                              (ATA)
        Set display OPTION for vendor Attribute N (see man page)

  -F TYPE, --firmwarebug=TYPE                                           (ATA)
        Use firmware bug workaround:
        none, nologdir, samsung, samsung2, samsung3, xerrorlba, swapid

  -P TYPE, --presets=TYPE                                               (ATA)
        Drive-specific presets: use, ignore, show, showall

  -B [+]FILE, --drivedb=[+]FILE                                         (ATA)
        Read and replace [add] drive database from FILE
        [default is +/usr/etc/smart_drivedb.h
         and then    /usr/share/smartmontools/drivedb.h]

========================================== DEVICE SELF-TEST OPTIONS =====

  -t TEST, --test=TEST
        Run test. TEST: offline, short, long, conveyance, force, vendor,N,
                        select,M-N, pending,N, afterselect,[on|off]

  -C, --captive
        Do test in captive mode (along with -t)

  -X, --abort
        Abort any non-captive test on device

================================================= SMARTCTL EXAMPLES =====

  smartctl --all /dev/hda                     (Prints all SMART information)
```

```
  smartctl --smart=on --offlineauto=on --saveauto=on /dev/hda
                                              (Enables SMART on first disk)

  smartctl --test=long /dev/hda          (Executes extended disk self-test)

  smartctl --attributes --log=selftest --qu
MODE, --nocheck=MODE (ATA) No check if: never, sleep, standby, idle (see man page)
============== DEVICE FEATURE ENABLE/DISABLE COMMANDS =====
```

bonnie++

bonnie++ is a test suite for storage devices that runs more comprehensive tests than the standard file system tests using the storagetool. You can run bonnie++ outside of the storagetool, but for logging purposes, use bonnie++ within storagetool.

```
root@dell-diag-os:/opt/dell/diag/bin# bonnie++
You must use the "-u" switch when running as root.
usage: bonnie++ [-d scratch-dir] [-s size(Mb)[:chunk-size(b)]]
                [-n number-to-stat[:max-size[:min-size][:num-directories]]]
                [-m machine-name]
                [-r ram-size-in-Mb]
                [-x number-of-tests] [-u uid-to-use:gid-to-use] [-g gid-to-use]
                [-q] [-f] [-b] [-p processes | -y]

Version: 1.03
root@dell-diag-os:/opt/dell/diag/bin#
```

temptool

The temptool reads from the temperature devices and reports back the temperatures.

The temperature sensors on the board are commonly connected through i2c busses. The configuration files specify the type of the device, the sensor name, the instance in that device, its location on the board, and the thresholds for reporting low, normal, and critical temperatures. To gather the information from the devices and report the values, the temptool uses the i2ctool.

Tests

The tool retrieves the data from the devices and validates that the temperatures are within the acceptable range.

CLI options

NOTE: Before using any commands, you must set the MUX settings to select the bus segments the temperature sensors are on.

```
DellEmc Diag - Temperature Tool
version 1.4, x.xx.x.x-x
build, 2017/05/23,

Syntax: temptool <option>
  Show the help-text:=
       temptool --h                                          (or)
       temptool -h
  Test the pre-programmed configuration:=
       temptool --test --config=<config_file> [--lpc]        (or)
       temptool -t -f <config_file> [-l]
  Execute repeatedly command by count:=
       temptool --iteration=max/<count> [option1] [option2]...(or)
       temptool -I max/<count> [option1] [option2]...
  Show the current temperature-device values:=
       temptool --show --config=<config_file> [--lpc]        (or)
       temptool -x -f <config_file> [-l]

Usage:=
   -h, --h           Show the help text
```

```
   -t, --test        Test using the pre-programmed configuration or use supplied config
   -x, --show        Show operation
   -f, --config=     To specify the location of the config file e.g. /etc/dn/diag/<file_name>
   -I, --iteration=  Iteration command execution
   -q, --lpc         Use LPC interface for reading temperature
                     LPC option MUST be used with show/test flags
```

- `test` — Tests the sensors to make sure they are within the acceptable range.
- `show` — Shows the current temperature values.

Output

test output

```
root@dell-diag-os:/opt/dell/diag/bin# temptool --test --lpc
Testing Temp sensor  devices:
Temperature Sensor 1 ..................... Passed
Temperature Sensor 2 ..................... Passed
Temperature Sensor 3 ..................... Passed
Temperature Sensor 4 ..................... Passed
Temperature Sensor 5 ..................... Passed
Temperature Sensor 6 ..................... Passed
Temperature Sensor 7 ..................... Passed
Temperature Sensor 8 ..................... Passed
Temperature Sensor 9 ..................... Passed
Temp Sensors: Overall test results ----- >>> Passed
root@dell-diag-os:/opt/dell/diag/bin#

root@dell-diag-os:/opt/dell/diag/bin# temptool --show --lpc
Temperature Sensor 1 temperature value is 30.3 C
Temperature Sensor 2 temperature value is 23.1 C
Temperature Sensor 3 temperature value is 22.2 C
Temperature Sensor 4 temperature value is 26.0 C
Temperature Sensor 5 temperature value is 21.8 C
Temperature Sensor 6 temperature value is 22.0 C
Temperature Sensor 7 temperature value is 23.5 C
Temperature Sensor 8 temperature value is 31.0 C
Temperature Sensor 9 temperature value is 42.0 C
root@dell-diag-os:/opt/dell/diag/bin#
```

updatetool

Use this command to update CPLD:

```
# update --dev=CPLD --index=1 --update --file= </mnt/media/cpld_image.vme>
```

```
root@dellemc-diag-os:~# updatetool --dev=CPLD --index=1 --update --file= /mnt/media/xxxx-xxxxx-xx_xxx_xxx.vme
 00
Disable device protect

Disable CPLD protect operation success, wait HW reset
Write image to CPLD
INFO: Yafu INI Configuration File not found... Default options will not be applied...

Creating IPMI session via USB...Done

------------------------------------------------
YAFUFlash - Firmware Upgrade Utility (Version 5.0.0)
------------------------------------------------
(C)Copyright 2016, American Megatrends Inc.
Beginning CPLD Update...
Uploading Image : 100%... done
Flashing Firmware Image : 100%... done
Verifying Firmware Image : 100%... done
write CPLD image success
```

```
Enable device protect

Update CPLD image success
root@dellemc-diag-os:~# shutdown -h now
[ xxx.xxxxxx] reboot: Power down
```

(UNPLUGGED POWER CABLES)

```
BIOS Boot Selector for VEP4600
Primary BIOS Version x.xx.x.x-xx

CPLD Version:x.x
CPLD Reset Source=0x44

root@dellemc-diag-os:~# updatetool --dev=ALL --device_version
BIOS version:
x.xx.x.x-xx

CPLD version:
CPLD_VERSION : offset 0x600 = 0x9
7: 4 MAJOR_VER = x
3: 0 MINOR_VER = x

MAIN-BMC version:
x.xx

BACKUP-BMC version:
x.xx
```

NOTE: When the update is complete, you must unplug and replug power cables to update the version.

Diagnostic package

The diagnostic applications, libraries, and configurations are packaged in a debian package called `dn-diags-{PLATFORM}-{PACKAGE_VERSION}.deb`.

Executables are placed in `/opt/ngos/bin`, libraries are placed in `/opt/ngos/lib`, and configurations are placed in `/etc/dn/diag`. To install the package on the switch, use the `dpkg --install <package_name>` command.

6

Dell EMC support

The Dell EMC support site provides documents and tools to help you effectively use Dell EMC equipment and mitigate network outages. Through the support site you can obtain technical information, access software upgrades and patches, download available management software, and manage your open cases. The Dell EMC support site provides integrated, secure access to these services.

To access the Dell EMC support site, go to www.dell.com/support/. To display information in your language, scroll down to the bottom of the web page and select your country from the drop-down menu.

- To obtain product-specific information, enter the 7-character service tag, or the 11-digit express service code of your platform and click **submit**.

  To view the platform service tag or express service code, pull out the luggage tag on the upper-right side of the platform or retrieve it remotely using the `ipmitool -H <bmc ip address> -I lanplus -U <user name> -P <password> fru` command.
- To receive more technical support, click **Contact Us**. On the Contact Information web page, click **Technical Support**.

To access platform documentation, go to www.dell.com/manuals/.

To search for drivers and downloads, go to www.dell.com/drivers/.

To participate in Dell EMC community blogs and forums, go to www.dell.com/community.