

Open Networking Hardware Diagnostic Guide

N2200-ON and N3200-ON Series Switches

October 2019

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.

© 2019 Dell Inc. or its subsidiaries. All rights reserved. Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners.

1 About this guide.....	5
Related documents.....	5
2 Overview.....	6
Boot processes.....	6
POST.....	6
Capture support data from ONIE.....	6
Change default grub boot entry for the N2200-ON and N3200-ON Series.....	7
View system information.....	7
3 ONIE installation instructions.....	9
ONIE installation.....	9
ONIE service discovery and OS installation.....	9
4 N2200-ON and N3200-ON Series instructions.....	11
Install ONIE from BIOS.....	11
ONIE UEFI-based installation using USB.....	11
ONIE-based license tool.....	16
5 Dell EMC DIAG OS for N2200-ON and N3200-ON Series.....	18
DIAG OS installation or update.....	18
Install or upgrade DIAG tools.....	19
Diagnostic test suite.....	19
View DIAG versions.....	19
View CPLD versions.....	19
6 Dell EMC DIAG-OS tools.....	20
cputool.....	20
edatool.....	22
eepromtool.....	29
ethtool.....	31
fantool.....	31
gpiotool.....	33
i2ctool.....	34
ledtool.....	39
lpctool.....	40
mentool.....	41
nputool.....	46
nvramtool.....	50
opticstool.....	52
pcitool.....	56
phytool.....	59
pltool.....	61

poetool.....	63
psutool.....	66
rtctool.....	68
smbiostool.....	69
storagetool.....	70
temptool.....	73
updatetool.....	74
Diagnostic package.....	77
7 Restore factory defaults.....	78
8 Dell EMC support.....	79

About this guide

Related documents

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

- *Dell EMC Warnings Guide*
- *Dell EMC Setup Placemat*
- *Dell EMC Installation Guide*
- *Dell EMC Release Notes*

For the most recent documentation, see Dell EMC support: www.dell.com/support.

Overview

When the diagnostic tools finish, your system displays pass or fail test results. If all tests pass, the diagnostic tools exit normally. If a test fails, each diagnostic tool offers a different result.

NOTE: For more information, see the README files that are associated with your platform.

NOTE: To download the Release Notes, go to www.dell.com/support.

The N2200-ON Series and N3200-ON Series switches use the power-on self-test (POST) diagnostic tool that automatically runs during the system power-on at the BIOS level. This tool tests for catastrophic hardware failures that prevent booting the system. The error code is saved in CMOS for the next boot. There is no physical alarm indication.

ONIE expansion for N2200-ON Series and N3200-ON Series

To view all the ONIE commands available, first select ONIE from the main menu. Then from the ONIE prompt, enter `onie-` and click **<tab>** twice.

```
ONIE:/ # onie- <TAB><TAB>
onie-boot-mode      onie-fwpkg          onie-syseeprom
onie-console        onie-nos-install    onie-sysinfo
onie-discovery-start onie-self-update    onie-uninstaller
onie-discovery-stop onie-support
```

Topics:

- [Boot processes](#)
- [POST](#)
- [Capture support data from ONIE](#)
- [View system information](#)

Boot processes

After the BIOS, for the N2200-ON Series and N3200-ON Series, POST runs to verify the devices required to boot to ONIE GRUB.

POST

POST diagnostics verify the system DRAM, DIMM, SPD, memory, RTC/NVRAM, and PCI devices. Test configuration parameters and test results are saved in NVRAM.

Capture support data from ONIE

1. Capture support data to the screen.

```
ONIE:/ # dmesg
```

2. Capture support data to the `onie-support.tar.bz2` gzip file.

ONIE-support creates the support file. To store the file, enter the location; for example, `ONIE:/# onie-support/tmp`.

```
ONIE:/ # onie-support <output_directory>
```

The ONIE support file includes the following:

- `kernel_cmdline`

- runtime-export-env
- runtime-process
- runtime-set-env
- log/messages
- log/onie.log

Output example

```
Success: Support tarball created: /tmp/onie-support.tar.bz2
```

Change default grub boot entry for the N2200-ON and N3200-ON Series

For the N2200-ON Series and N3200-ON Series, to view or set the default Boot mode, the `onie-boot-mode` command has two options `-l`, the default, and `-o`. The Grub boot default shows the current default entry.

View or set the default Grub boot entry.

```
ONIE:/ # onie-boot-mode [-o <onie_mode>]
```

The `-o` command options include:

- `install`—ONIE OS Installer mode
- `rescue`—ONIE Rescue mode
- `uninstall`—ONIE OS Uninstall mode
- `update`—ONIE Self-Update mode
- `embed`—ONIE Self-Update mode and Embed ONIE
- `none`—Uses System Default Boot mode. This mode uses the first ONIE boot menu entry.

The `-l` command option lists the current default entry—this is the default setting.

View system information

To view your system information; for example, the model, part number, serial number, and service tag, use the following commands:

1. Boot into ONIE.
2. Enter the `onie-syseeprom` command.

```
ONIE:/ # onie-syseeprom
TlvInfo Header:
  Id String:      TlvInfo
  Version:        1
  Total Length:  162
TLV Name          Code Len Value
-----
Part Number       0x22  6 0W1K08
Serial Number     0x23 20 CNOW1K08779316470002
Product Name      0x21  8 <platform>
Device Version    0x26  1 0
Label Revision    0x27  3 X00
Manufacture Date  0x25 19 04/08/2016 08:43:05
Manufacturer      0x2B  5 77931
Country Code      0x2C  2 CN
Vendor Extension  0xFD  1 0x00
MAC Addresses     0x2A  2 256
Service Tag       0x2F  7 2WCSG02
Vendor Name       0x2D  4 DELL
Diag Version      0x2E  6 01_010
Base MAC Address  0x24  6 34:17:EB:05:B4:00
Platform Name     0x28 26 x86_64-dell_<platform>_c2538-r0
ONIE Version      0x29  8 x.xx.x.x
CRC-32           0xFE  4 0x99415608
Checksum is valid.
ONIE:/ #
```

3. Enter the `onie-sysinfo -a` command.

```
ONIE:/ # onie-sysinfo -a
```

```
CN0W1K08779316470002 0W1K08 34:17:EB:05:B4:00 x.xx.x.x 674 dell_<platform>_c2538 0 x86_64-  
dell_<platform>_c2538-r0 x86_64 1 gpt 2016-09-21T10:01-0700 bcm  
ONIE:/ #
```


ONIE installation instructions

This section describes the different methods to install ONIE on your switch.

NOTE: After installing the networking operating software (NOS) and diagnostics operating system (DIAG-OS), if you boot into ONIE Install mode, ONIE assumes ownership of the system; ONIE Install mode is sticky. In this situation, ONIE stays in Install mode until NOS and DIAG-OS are successfully installed again. If you want to boot into ONIE for any reason other than installation, use Rescue mode or Update mode.

NOTE: To access ONIE, use the RJ45 console port or MicroUSB console port.

Topics:

- [ONIE installation](#)
- [ONIE service discovery and OS installation](#)

ONIE installation

The following steps describe how to load ONIE:

- Installing ONIE—these instructions use the universal serial bus (USB) method. To boot from a Linux USB, you must install the preprogrammed BIOS on your platform.
- ONIE operates using a 115200 baud rate. Ensure that any equipment that is attached to the serial port supports the required 115200 baud rate.

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 that is shown are for figure purpose only. Use your system-applicable values.

ONIE service discovery and OS installation

ONIE attempts to locate the installer through several discovery methods.

To download and run an installer, the ONIE Service Discovery feature follows these steps in order and uses the first successful method found:

1. Pass from the boot loader.
2. Search locally attached storage devices for one of the ONIE default installer filenames—for example, USB.
3. Discover the URLs from DHCPv4.
4. Report discovered URLs based on the DHCPv4 responses.
5. Query to the IPv6 link-local neighbors using HTTP for an installer.
6. Start TFTP waterfall—from the DHCPv4 option 66.

If none of the ONIE Service Discovery methods are successful, you can disable this using the `onie-discovery-stop` command.

You can install an operating system manually from HTTP, USB, FTP, or TFTP using the `onie-nos-install <URL>` command.

NOTE: If you have a recovery USB plugged into your system, you must remove it before using the `onie-nos-install <URL>` command.

The ONIE Install environment uses DHCP to assign an IP address to the management interface, `eth0`. If that fails, it uses the link-local IPv4 address 169.254.209.190/16.

To display the IP address, use the `ifconfig eth0` command, as shown:

```
ONIE:/ # ifconfig eth0
eth0 Link encap:Ethernet HWaddr 90:B1:1C:F4:9C:76
```

```
inet addr:x.x.x.x Bcast:x.x.x.x Mask:x.x.x.x
inet6 addr: fe80::92b1:1cff:fef4:9c76/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:18 errors:0 dropped:0 overruns:0 frame:0
TX packets:24 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000
RX bytes:1152 (1.1 KiB) TX bytes:6864 (6.7 KiB)
Interrupt:21 Memory:ff300000-ff320000
```

To assign an IP address to the management interface, eth0, and verify network connectivity, use the `ifconfig eth0 <ip address>` command, as shown:

```
ONIE:/ # ifconfig eth0 x.x.x.x netmask x.x.x.x UP
```

Then set speed on management interface as below

```
ONIE:/ # ethtool -s eth0 speed 100 duplex full
Verify the network connection with ping.
ONIE:/ # ping x.x.x.x
PING x.x.x.x (x.x.x.x): 56 data bytes
64 bytes from x.x.x.x: seq=0 ttl=62 time=1.357 ms
64 bytes from x.x.x.x: seq=1 ttl=62 time=0.577 ms
^C
```

N2200-ON and N3200-ON Series instructions

This section applies to the N2200-ON and N3200-ON Series platforms only.

Topics:

- [Install ONIE from BIOS](#)
- [ONIE UEFI-based installation using USB](#)
- [ONIE-based license tool](#)

Install ONIE from BIOS

Install ONIE from the BIOS using the media (usb) boot—the ONIE installer USB.

Pre-requisites

The BIOS running on your system must meet the following requirements:

- Allows a change to the boot order so the system can boot from media (USB).
- Allows a baud-rate change. This feature is optional; you do not need it if your BIOS is running at 115200 baud rate. The default baud rate for ONIE is 115200.

⚠ CAUTION:

- **These procedures are for Intel x86-based targets only.**
- **Check the console (UART-0/1) used on the target. Connect the serial port at 115200 baud rate and check that message display on the console.**
- **The log messages in this guide are subject to change.**

i NOTE: The following procedure is generic and does not list a particular target. The ONIE images are specified using the `<platform>_<cpu>` notation. For example, the ONIE media (usb) iso image is `onie-recovery-x86_64-dell_<platform>_<cpu>-r0.iso`.

ONIE UEFI-based installation using USB

The following steps describe how to create a bootable unified extensible firmware interface (UEFI) ONIE-based USB to install ONIE using Embed mode:

To install ONIE UEFI on your system, use any existing ONIE-based system to make an ONIE UEFI-based bootable USB. To make a bootable USB, use the ONIE ISO file.

1. Boot the ONIE target in ONIE Rescue mode.

Use ONIE Rescue mode to make the ONIE UEFI-bootable USB.

To select which entry is selected, use the up and down arrow keys. Press Enter to select an operating software-selected OS or enter e to edit the commands before booting. Enter c for a command line. The selected entry, displaying *, runs automatically in the operating system.

```
GNU GRUB  version 2.02~beta2+e4a1fe391
```

```
+-----+
|ONIE: Install OS          |
|*ONIE: Rescue            |
|ONIE: Uninstall OS      |
|ONIE: Update ONIE       |
|ONIE: Embed ONIE        |
|EDA-DIAG                 |
|                          |
|                          |
|                          |
```

```
|
+-----+
```

2. Confirm that your system can reach the network.
3. Copy the ONIE ISO image to the solid-state drive (SSD) of the ONIE target.

```
ONIE:/ # wget --quiet http://xx.xx.x.xxx/tftpboot/users/<name>/onie-recovery-x86_64-dell_<platform>_c2538-r0.iso
```

To copy the image, you can use SCP, TFTP, or WGET (ftp/http).

```
scp username@xx.xx.xxx.xxx:/tftpboot/onie-recovery-x86_64-dell_<platform>_c2538-r0.iso .
```

4. Confirm that the ISO file copied to the SDD over the network.

```
ONIE:/ # ls -l
...
-rw-r--r-- 1 root 0 39780352 Apr 10 11:55 onie-recovery-x86_64-dell_<platform>_c2538-r0.iso
...
```

5. Insert a blank USB in the ONIE target USB slot. Verify the USB block device using the ONIE logs.

```
Info: eth0: Checking link... scsi 6:0:0:0: Direct-Access Kingston DataTraveler 2.0 1.00
PQ: 0 ANSI: 4
sd 6:0:0:0: [sdb] 15148608 512-byte logical blocks: (7.75 GB/7.22 GiB)
sd 6:0:0:0: [sdb] Write Protect is off
sd 6:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sd 6:0:0:0: [sdb] Attached SCSI removable disk
```

The logs show that the USB device is present: /dev/sdb.

You can also check /sys/block.

```
ONIE:/ # cd /sys/block/sdb
ONIE:/sys/block/sdb # ls -l
-r--r--r-- 1 root 0 4096 Apr 10 13:12 alignment_offset
lrwxrwxrwx 1 root 0 0 Apr 10 13:12 bdi -> ../../devices/virtual/bdi/8:16
-r--r--r-- 1 root 0 4096 Apr 10 13:12 capability
-r--r--r-- 1 root 0 4096 Apr 10 13:12 dev
lrwxrwxrwx 1 root 0 0 Apr 10 13:12 device -> ../../devices/pci0000:00/0000:00:16.0/
usb1/1-1/1-1.1/1-1.1.1/1-1.1.1:1.0/host6/target6:0:0/6:0:0:0
...
ONIE:/sys/block/sdb # cd
```

6. Copy the ISO image to the USB using the dd command.

```
ONIE:/ # dd if=./onie-recovery-x86_64-dell_<platform>_c2538-r0.iso of=/dev/sdb bs=10M
3+1 records in
3+1 records out
39780352 bytes (37.9MB) copied, 6.890503 seconds, 5.5MB/s
ONIE:/ #
```

7. Move the USB from the ONIE target—the system with ONIE—to the USB slot in your switch—the system without ONIE.
8. Turn-on your system and enter the **BIOS setup** menu by pressing the **Del** key when the BIOS message displays.
If you already powered on your system, reboot the system and enter the **BIOS setup** menu by pressing the **Del** key.
9. In the **BIOS Boot** menu, select **UEFI USB**. Select **Save and Exit**.

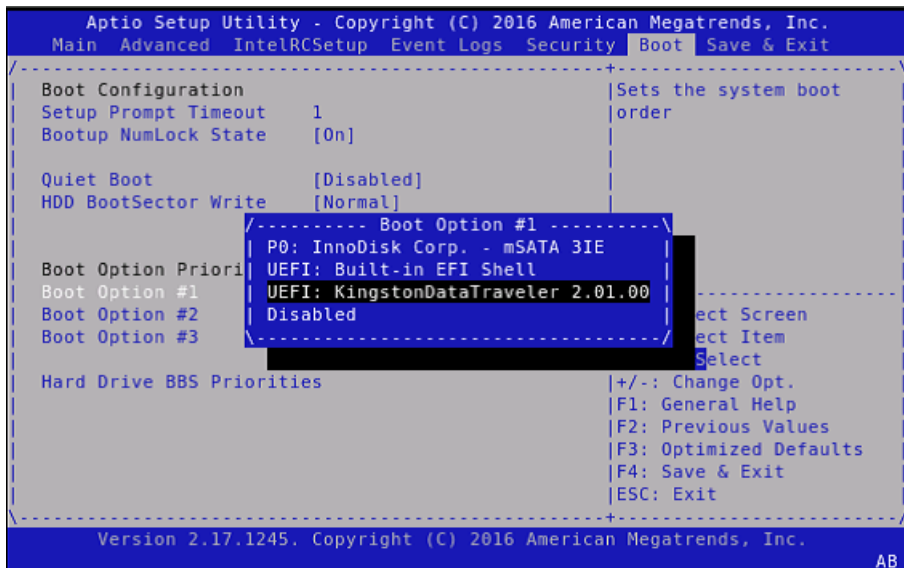


Figure 1. Setup utility

After the system exits the **BIOS Boot** menu, the system boots with the ONIE USB and presents the following menu:



Figure 2. Embed ONIE menu

10. Select the Embed ONIE option.

This step installs the UEFI ONIE version 3.26.1.1 on system. Any previous installation is removed.

Do not press any key during the Embed ONIE installation.

The following are the Embed ONIE logs:

```

ONIE: Embedding ONIE ...
Platform : x86_64-dell_<platform>_c2538-r0
Version : x.xx.x.x
Build Date: 2016-04-26T09:14-0700
[ 4.066378] dummy-irq: no IRQ given. Use irq=N
[ 14.296290] esas2r: driver will not be loaded because no ATTO esas2r devices were found
[ 14.463587] mtdoops: mtd device (mtddev=name/number) must be supplied
[ 16.328319] i8042: No controller found
[ 16.397853] fmc_write_eeprom fake-design-for-testing-f001: fmc_write_eeprom: no busid
passed, refusing all cards
[ 16.568122] intel_rapl: driver does not support CPU family 6 model 77
Info: Mounting kernel filesystems... done.
Info: Mounting ONIE-BOOT on /mnt/onie-boot ...
Info: Using eth0 MAC address: 4c:76:25:f4:7c:80
Info: eth0: Checking link... [ 18.571495] scsi 6:0:0:0: Direct-Access Kingston
DataTraveler 2.0 1.00 PQ: 0 ANSI: 4

```

```

[ 18.707185] sd 6:0:0:0: Attached scsi generic sgl type 0
[ 18.707703] sd 6:0:0:0: [sdb] 15148608 512-byte logical blocks: (7.75 GB/7.22 GiB)
[ 18.796392] sd 6:0:0:0: [sdb] Write Protect is off
[ 18.797033] sd 6:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support
DPO or FUA
[ 19.159563] sd 6:0:0:0: [sdb] Attached SCSI removable disk
up.
Info: Trying DHCPv4 on interface: eth0
ONIE: Using DHCPv4 addr: eth0: 1[ 20.053045] random: dropbearkey urandom read with 94 bits
of entropy available
x.xx.xxx.xx / xxx.xxx.xxx.x
Starting: dropbear ssh daemon... done.
Starting: telnetd... done.
discover: ONIE embed mode detected. Running updater.
Starting: discover... done.
Please press Enter to activate this console. Info: eth0: Checking link... up.
Info: Trying DHCPv4 on interface: eth0
ONIE: Using DHCPv4 addr: eth0: x.xx.xxx.xx / xxx.xxx.xxx.x
ONIE: Starting ONIE Service Discovery
Info: Found static url: file:///lib/onie/onie-updater
[ 29.744855] random: nonblocking pool is initialized
ONIE: Executing installer: file:///lib/onie/onie-updater
Verifying image checksum ... OK.
Preparing image archive ... OK.
ONIE: Version : x.xx.x.x
ONIE: Architecture : x86_64
ONIE: Machine : dell_<platform>_c2538
ONIE: Machine Rev : 0
ONIE: Config Version: 1
Installing ONIE on: /dev/sda
/proc/devices: No entry for device-mapper found
/proc/devices: No entry for device-mapper found
ONIE: Success: Firmware update URL: file:///lib/onie/onie-updater
ONIE: Success: Firmware update version: x.xx.x.x
ONIE: Rebooting...
discover: ONIE embed mode detected.
Stopping: discover...start-stop-daemon: warning: killing process 1441: No such process
Stopping: dropbear ssh daemon... done.
Stopping: telnetd... done.

Stopping: syslogd... done.
Info: Unmounting kernel filesystems
The system is going down NOW!
Sent SIGTERM to all processes
Sent SIGKILL to all processes
Requesting system reboot

```

After the Embed-ONIE installation completes, the system bootups and presents the **ONIE** menu.



Figure 3. ONIE install menu

The system comes up in ONIE Install mode by default, as shown:

```
ONIE: OS Install Mode ...
Version : x.xx.x.x
Build Date: 2016-04-26T09:14-0700
ONIE: OS Install Mode ...
Version : x.xx.x.x
Build Date: 2016-04-26T09:14-0700
[ 4.759116] dummy-irq: no IRQ given. Use irq=N

[ 4.835970] esas2r: driver will not be loaded because no ATTO esas2r
devices were found
[ 5.003050] mtdoops: mtd device (mtddev=name/number) must be supplied
[ 6.867708] i8042: No controller found
[ 6.937375] fmc_write_eeprom fake-design-for-testing-f001:
fmc_write_eeprom: no busid passed, refusing all cards
[ 7.107669] intel_rapl: driver does not support CPU family 6 model 77
Info: Mounting kernel filesystems... done.
Info: Mounting ONIE-BOOT on /mnt/onie-boot ...
[ 8.018377] random: fscck urandom read with 73 bits of entropy available
Info: Mounting EFI System on /boot/efi ...
Info: Using eth0 MAC address: 4c:76:25:f4:7c:80
Info: eth0: Checking link... [ 8.902787] scsi 6:0:0:0: Direct-Access
Kingston DataTraveler 2.0 1.00 PQ: 0 ANSI: 4
[ 9.038475] sd 6:0:0:0: Attached scsi generic sgl type 0
[ 9.038993] sd 6:0:0:0: [sdb] 15148608 512-byte logical blocks: (7.75
GB/7.22 GiB)
[ 9.253877] sd 6:0:0:0: [sdb] Write Protect is off
[ 9.254546] sd 6:0:0:0: [sdb] Write cache: disabled, read cache: enabled,
doesn't support DPO or FUA
[ 9.492124] sd 6:0:0:0: [sdb] Attached SCSI removable disk
up.
Info: Trying DHCPv4 on interface: eth0
ONIE: Using DHCPv4 addr: eth0: x.xx.xxx.xx / xxx.xxx.xxx.x
Starting: dropbear ssh daemon... done.
Starting: telnetd... done.
[ 11.789298] random: nonblocking pool is initialized
discover: installer mode detected. Running installer.
Starting: discover... done.
Please press Enter to activate this console. Info: eth0: Checking link... up.
Info: Trying DHCPv4 on interface: eth0
ONIE: Using DHCPv4 addr: eth0: x.xx.xxx.xx / xxx.xxx.xxx.x
ONIE: Starting ONIE Service Discovery
Info: Fetching
http://xx.xx.xxx.x/onie-installer-x86_64-dell_<platform>_c2538-r0 ...
Info: Fetching http://xx.xx.xxx.x/onie-installer-x86_64-dell_<platform>_c2538
...
Info: Fetching http://xx.xx.xxx.x/onie-installer-dell_<platform>_c2538 ...
Info: Fetching http://xx.xx.xxx.x/onie-installer-x86_64 ...
Info: Fetching http://xx.xx.xxx.x/onie-installer ...
Info: Fetching
http://xx.xx.xxx.x/onie-installer-x86_64-dell_<platform>_c2538-r0 ...
Info: Fetching http://xx.xx.xxx.x/onie-installer-x86_64-dell_<platform>_c2538
...
```

11. Stop ONIE Discovery mode.

```
ONIE:/ # onie-discovery-stop
The operation has completed successfully.
ONIE:/ #
```

12. Verify the ONIE Linux kernel version and partition layout.

This step verifies that you are running the correct kernel in ONIE as the kernel is separate from the ONIE environment.

```
ONIE:/ # uname -a
Linux onie 4.1.28-onie+ #1 SMP Wed Sep 7 14:38:43 PDT 2016 x86_64 GNU/Linux
ONIE:/ # sgdisk -p /dev/sda
Disk /dev/sda: 31277232 sectors, 14.9 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 763E53FF-B894-40FD-B0F9-FBAE2ED4B0B5
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 31277198
```

```
Partitions will be aligned on 2048-sector boundaries
Total free space is 30490733 sectors (14.5 GiB)
Number Start (sector) End (sector) Size Code Name
1 2048 526335 256.0 MiB EF00 EFI System
2 526336 788479 128.0 MiB 3000 ONIE-BOOT
ONIE:/ #
```

- Verify that `efibootmgr` runs and displays the valid boot options.

```
ONIE:/ # efibootmgr
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0000,0006,0001,0003
Boot0000* ONIE: Open Network Install Environment
Boot0001* Hard Drive
Boot0003* UEFI: Built-in EFI Shell
Boot0006* UEFI: KingstonDataTraveler 2.01.00 14
```

ONIE-based license tool

The N2200-ON Series and N3200-ON Series platforms support the ONIE-based license tool. Use this tool to install or uninstall the OS6 license using a valid license (.xml) file in ONIE Rescue mode.

The following conditions apply:

- If the networking operating system (NOS) to install is OS6 AND the switch is a no-OS platform, you must install a license file.
 - If the NOS is anything other than OS6 OR the switch is a factory-installed OS6 platform, you do not need a license and the NOS operates as expected.
- Get a valid license file for your platform from the Dell EMC support site at www.dell.com/support.
The license file is in .xml format and contains an X509 security certificate. Each license file is valid for a particular service-tag.
 - Get the latest ONIE license tool from the Dell EMC support site at www.dell.com/support.
 - Boot into ONIE Rescue mode.
 - Copy the .xml license file and ONIE license tool to your platform.
The ONIE license tool, `license_install.run`, is a self-extracted binary.
 - Ensure that the tool has `execute(x)` permission.

```
ONIE:/ # ls -ltr license_install.run
-rwxr-xr-x 1 root 0 15638815 May 9 22:57 license_install.run
ONIE:/ #
```

If the tool does not have `execute(x)` permission, add `execute(x)` permission.

```
ONIE:/ # chmod +x license_install.run
```

- Install the ONIE license tool.

```
ONIE:/ # ./license_install.run
/dev/console
Verifying archive integrity... 100% 100% All good.
Uncompressing LICENSE INSTALLER 100%
.....
Done
```

- Run the license command to confirm the license tool installed correctly.

```
ONIE:/ # license
Version 1.0
Error: wrong number of arguments.

CLI option description.
Install a new license or update existing license.
-i <xml_license_file>
Un-install the license.
-u
ONIE:/ #
```


8. Install the valid .xml license file.

```
ONIE:/ # license -i <Service-Tag>-<XML_LICENSE_FILE>-License.xml
Version 1.0
license_file B3NFXC2-NOSOS6-License.xml

CLI option description.
Install a new license or update existing license.
-i <xml_license_file>
Un-install the license.
-u
ONIE:/ #
```

To uninstall a license, use the `license -u` command.

```
ONIE:/ # license -u
Version 1.0
ONIE:/ #
```

Dell EMC DIAG OS for N2200-ON and N3200-ON Series

These sections describe the Dell EMC diagnostics.

Topics:

- [DIAG OS installation or update](#)
- [Diagnostic test suite](#)
- [View DIAG versions](#)
- [View CPLD versions](#)

DIAG OS installation or update

Load or update the DIAG-OS—the diag installer image—using the `onie-nos-install` command. The DIAG-OS installer runs in two modes: Update mode or Install mode.

- In Update mode, the DIAG-OS updates the existing DIAG-OS and boots back to ONIE.
- In Install mode, the DIAG-OS erases the existing DIAG-OS and loads the new DIAG-OS.

NOTE: If you have a recovery USB plugged into your system, remove it before using the `onie-nos-install` command.

NOTE: Before you begin, go to www.dell.com/support and download the diagnostic package.

1. Enter the `onie-discovery-stop` command to stop ONIE Discovery mode.
2. Assign an IP address to the management interface and verify the network connectivity.

```
ONIE:/ # ifconfig eth0 xx.xx.xx.xx netmask xxx.xxx.x.x up
ONIE:/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 34:17:EB:05:B4:00
          inet addr:xx.xx.xx.xx  Bcast:xx.xx.xxx.xxx  Mask:xxx.xxx.x.x
          inet6 addr: fe80::3617:ebff:fe05:b400/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5118 (4.9 KiB)  TX bytes:7104 (6.9 KiB)
          Memory:dff40000-dff5ffff
```

3. Upgrade the DIAG Installer.

NOTE: In Install mode, the DIAG-OS installation removes any existing NOS and DIAG-OS partition. If you do not create file `/tmp/diag_os_install_mode`, the DIAG-OS installs in Upgrade mode. In this case, the installation process does NOT touch any existing NOS.

4. Start diagnostics.

To start the diagnostics, use the `EDA-DIAG` option from the GRUB menu.

- a) Boot into the EDA DIAGs.
- b) Log in as `root`.
Password: `calvin`.
- c) Install the EDA-DIAG tools package.

NOTE: To return to your networking operating software, enter the `reboot` command.

Install or upgrade DIAG tools

To install or upgrade the DIAGs in the DIAGs OS, use the `dpkg --install dn-diags-<platform>-DiagOS-<version>-<date>.deb` command.

NOTE: If DIAG tools are installed on your switch, you must first remove them using the `dpkg -P dn-diags-evans-on.deb` command before re-installing or upgrading the DIAG tools.

```
root@dell-diag-os:~#dpkg --install dn-diags-<platform>-DiagOS-<version>-<date>.deb
Selecting previously unselected package dn-diags-<platform>.deb.
(Reading database ... 18873 files and directories currently installed.)
Preparing to unpack dn-diags-<platform>-DiagOS-<version>-<date>.deb ...
Unpacking dn-diags-<platform>.deb (1.10) ...
Setting up dn-diags-<platform>.deb (1.10) ...
root@dell-diag-os:~#
```

Diagnostic test suite

After the system boots up, select the EDA-DIAG option to run the diagnostic test suite.

To select which entry is highlighted, use the up and down arrow keys. Press Enter to select an operating software-selected OS or enter e to edit the commands before booting. Enter c for a command line. The highlighted entry, displaying *, executes automatically in the operating system.

```
GNU GRUB  version 2.02~beta2+e4a1fe391

+-----+
|ONIE: Install OS          |
|ONIE: Rescue              |
|ONIE: Uninstall OS       |
|ONIE: Update ONIE        |
|ONIE: Embed ONIE         |
|*EDA-DIAG                 |
|                           |
|                           |
|                           |
+-----+
```

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@dellemc-diag-os:~# dpkg -l | grep dn-diags
ii dn-diags-<platform>-on.deb 3.xx.4.1-x amd64 Dell Networking Diagnostics
```

View CPLD versions

To view CPLD data, including the fan status, PSU status, current programmed version, and image packed version, use the `cpdupgradetool` or `updatetool` command at the prompt.

For the `updatetool` command:

```
root@dellemc-diag-od~#updatetool --device_version --dev=CPU_CPLD
CPU_CPLD version:
System CPLD Version : offset 0x00 = 0xc
7: 4 Major Revision = 0
3: 0 Minor Revision = c
Scratch Register : offset 0x01 = 0x0
```

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 `roots` to support the tools and drivers.

NOTE: By default, the system's I/O modules are down. Power up the I/O modules or the Opticstool and NPUtil reports failures. For information about how to power up the I/O modules, see the *Dell EMC Installation Guide* for your system at www.dell.com/support.

Topics:

- `cputool`
- `edatool`
- `epromtool`
- `ethtool`
- `fantool`
- `gpiotool`
- `i2ctool`
- `ledtool`
- `lpctool`
- `memtool`
- `nputool`
- `nvrantool`
- `opticstool`
- `pcitool`
- `phytool`
- `pltool`
- `poetool`
- `psutool`
- `rtctool`
- `smbiostool`
- `storagetool`
- `temptool`
- `updatetool`
- Diagnostic package

`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 2.0, x.xx.x.x-x
build, 2019/07/03,

Syntax: cputool <option>
Show the help-text:=
  cputool --help                                     (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]
Display the CPU info:=
  cputool --cpuinfo                                 (or)
  cputool -C
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 register in LPC bus:=
  cputool --readlpc --reg=<reg> --size=<size>         (or)
  cputool -d -R <reg> -Z <size>
Write the specified register 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
  -C, --cpuinfo          CPU 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

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

Output

```
root@dellemc-diag-os:~# cputool -h
DellEmc Diag - Cpu Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

Syntax: cputool <option>
Show the help-text:=
  cputool --help                                     (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]
```

```

Display the CPU info:=
  cputool --cpuinfo                                     (or)
  cputool -C
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 register in LPC bus:=
  cputool --readlpc --reg=<reg> --size=<size>           (or)
  cputool -d -R <reg> -Z <size>
Write the specified register 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
  -C, --cpuinfo          CPU 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
root@dellemc-diag-os:~#
root@dellemc-diag-os:~# cputool --x86info
x86info v1.30. Dave Jones 2001-2011
Feedback to <davej@redhat.com>.

Found 2 identical CPUs
Extended Family: 0 Extended Model: 5 Family: 6 Model: 95 Stepping: 1
Type: 0 (Original OEM)
CPU Model (x86info's best guess): Unknown model.
Processor name string (BIOS programmed): Intel(R) Atom(TM) CPU C3338 @ 1.50GHz

Total processor threads: 2
This system has 1 single-core processor with hyper-threading (2 threads per core) running at
an estimated 1.50GHz

```

edatool

The diagnostics tools include `edatool`. To test the basic functionality of the system, use the `edatool`.

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. 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

```

root@dellemc-diag-os:~# edatool --sysinfo
Gathering System Data ... Please Wait ....

Software Info:
  Diag SW Version      : x.xx.x.x-x

```

```
Diag SW Build Date   : 2019/07/03
DiagOS Version       : x.xx.x.xx-x
Linux Version        : x.x.xx
SDK Version          : sdk-x.x.xx
Bios Version         : x.xx.x.x-x
Bios Build Date      : 06/18/2019
```

Physical Memory Information:

```
Maximum Capacity    : 3997100 kB
Number of Devices    : 1
Device Size         : 4096 MB
Error Correction Type : ECC: yes
Device Frequency     : 2400 Mhz
```

Manufacturing Information:

```
Serial Number (PPID) : TW0WYGRVDNT0091V0011
Device Version       : 1
Label Revision       : X00
Service Tag          : 1GPFXC2
Express Service Code : 3186974594
Part Number          : 0WYGRV
```

root@dellemc-diag-os:~#

Output

root@dellemc-diag-os:~# edatool

```
*****
* Diagnostics Application *
*****
DellEmc Diag edatool version x.x, package x.xx.x.x-x 2019/07/03
DellEmc Diag cputool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag eepromtool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag fantool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag gpiotool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag i2ctool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag ledtool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag lpctool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag memtool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag nputool - version x.x sdk-x.x.xx package x.xx.x.x-x 2019/07/03
DellEmc Diag nvramtool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag optictool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag pcitool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag phytool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag pltool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag psutool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag poetool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag rtctool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag smbiostool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag storagetool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag temptool - version x.x package x.xx.x.x-x 2019/07/03
DellEmc Diag UPDATETOOL - version x.x package x.xx.x.x-x 2019/07/03
Testing PCI devices:
+ Checking PCI 00:00.0, ID=19808086 ..... Passed
+ Checking PCI 00:04.0, ID=19a18086 ..... Passed
+ Checking PCI 00:05.0, ID=19a28086 ..... Passed
+ Checking PCI 00:09.0, ID=19a48086 ..... Passed
+ Checking PCI 00:0b.0, ID=19a68086 ..... Passed
+ Checking PCI 00:12.0, ID=19ac8086 ..... Passed
+ Checking PCI 00:14.0, ID=19c28086 ..... Passed
+ Checking PCI 00:15.0, ID=19d08086 ..... Passed
+ Checking PCI 00:18.0, ID=19d38086 ..... Passed
+ Checking PCI 00:1c.0, ID=19db8086 ..... Passed
+ Checking PCI 00:1f.0, ID=19dc8086 ..... Passed
+ Checking PCI 00:1f.2, ID=19de8086 ..... Passed
+ Checking PCI 00:1f.4, ID=19df8086 ..... Passed
+ Checking PCI 00:1f.5, ID=19e08086 ..... Passed
+ Checking PCI 01:00.0, ID=b77114e4 ..... Passed
+ Checking PCI 02:00.0, ID=15338086 ..... Passed
PCI devices: Overall test results----->>>> PASSED
Testing I2C devices:
```

```

Checking I2C devices on bus 0:
+ Checking CPU board Thermal Sensor_TMP75AI 0x48 ..... Passed
+ Checking SPD EEPROM_M34E04 0x50 ..... Passed
Checking I2C devices on bus 1:
+ Checking CPU CPLD 0x31 ..... Passed
+ Checking SYSTEM CPLD 0x32 ..... Passed
+ Checking I2C Switch 0x71 ..... Passed
+ Checking Main Board ID 0x50 ..... Passed
+ Checking IR3595 1 0x70 ..... Passed
+ Checking IR3595 2 0x72 ..... Passed
+ Checking FAN Tray 01 EEPROM 0x50 ..... Passed
+ Checking FAN Tray 02 EEPROM 0x50 ..... Passed
+ Checking FAN Tray 03 EEPROM 0x50 ..... Passed
+ Checking PSU1 EEPROM 0x56 ..... Passed
+ Checking PSU1 FAN CONTROL 0x5e ..... Passed
+ Checking FAN Controller_EMCC2305 0x2c ..... Passed
+ Checking Thermal Sensor 1 0x49 ..... Passed
+ Checking Thermal Sensor 2 0x4a ..... Passed
+ Checking Thermal Sensor 3 0x4b ..... Passed
+ Checking Thermal Sensor 4 0x4c ..... Passed
+ Checking Thermal Sensor 5 0x4f ..... Passed
I2C Devices: Overall test results----- >>>> PASSED
Testing Temp sensor devices:
+ Checking [CPU board Thermal Sensor_TMP75AI] = 32.0 C ..... Passed
+ Checking [Temp Sensor 1] = 29.0 C ..... Passed
+ Checking [Temp Sensor 2] = 27.0 C ..... Passed
+ Checking [Temp Sensor 3] = 26.0 C ..... Passed
+ Checking [Temp Sensor 4] = 33.0 C ..... Passed
+ Checking [Temp Sensor 5] = 30.0 C ..... Passed
Temp Sensors: Overall test results----- >>>> PASSED
PL Tool test:

CPU CPLD: CPU CPLD General Purpose Register Reg Addr: 0x2 .....Passed
System CPLD: System CPLD General Purpose Reg Reg Addr: 0x2 .....Passed
PL Tool: Overall test results----- >>>> PASSED
Fan Controller Short test: Overall test results----- >>>> PASSED
Fan 1 speed is 12098 RPM
Fan 2 speed is 12173 RPM
Fan 3 speed is 11988 RPM

Power Supply Test all
DELL PSU FAN Speed 5408 RPM
DELL PSU Current IN 0.5(A)
DELL PSU Current OUT 1.4(A)
DELL PSU Voltage IN 206.0(V)
DELL PSU Voltage Out 56.5(V)
DELL PSU Power IN 89.2(W)
DELL PSU Power Out 80.4(W)
DELL PSU Temperature 34.3(C)
PSU-1 ..... Passed
PSU-2 is not present

```


Power Supply Test: Overall test results----- >>>> PASSED

Show Optics in System

Port #	Name	Status	Type	Part Number	Rev	Serial Number
49	SFP+ 1	REMOVED				
50	SFP+ 2	REMOVED				
51	SFP+ 3	REMOVED				
52	SFP+ 4	REMOVED				

Error reading at offset 0x00

ERROR: i2c_read_tool[738]: ERROR: dev=0x50 adrs=0x00 (6): Not able to read all 256 bytes, read 0

53 QSFP28 1 PRESENTERROR: i2ctool_get[1055]: ERROR: FAILED <<<---

Error Reading from I2C

Error reading at offset 0x00

ERROR: i2c_read_tool[738]: ERROR: dev=0x50 adrs=0x00 (6): Not able to read all 256 bytes, read 0

ERROR: i2ctool_get[1055]: ERROR: FAILED <<<---

Error Reading from I2C

ERROR: get_optics_reg[3904]: ERROR: /dev/i2c-1 0x50 0x0 256 Failed, rc = 65280
UnknownNot supported optic type 0

Error reading at offset 0x00

ERROR: i2c_read_tool[738]: ERROR: dev=0x50 adrs=0x00 (6): Not able to read all 256 bytes, read 0

54 QSFP28 2 PRESENTERROR: i2ctool_get[1055]: ERROR: FAILED <<<---

Error Reading from I2C

Error reading at offset 0x00

ERROR: i2c_read_tool[738]: ERROR: dev=0x50 adrs=0x00 (6): Not able to read all 256 bytes, read 0

ERROR: i2ctool_get[1055]: ERROR: FAILED <<<---

Error Reading from I2C

ERROR: get_optics_reg[3904]: ERROR: /dev/i2c-1 0x50 0x0 256 Failed, rc = 65280
UnknownNot supported optic type 0

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

MAC Addr Test: Overall test results----- >>>> PASSED

Mounted Filesystem Devices:

/dev/mmcblk0p2 / ext4

rmmmod: ERROR: Module linux_user_bde is not currently loaded

rmmmod: ERROR: Module linux_kernel_bde is not currently loaded

DMA pool size: 33554432

PCI unit 0: Dev 0xb771, Rev 0x01, Chip BCM56771_A0, Driver BCM56770_A0

sysconf_probe successful

global_sal_config successful

*** 1 BCM devices are detected

SOC unit 0 attached to PCI device BCM56771_A0

Loading M0 Firmware located at /etc/dn/diag/linkscan_led_fw.bin

Loading M0 Firmware located at /etc/dn/diag/custom_led.bin

Adr=0x47: FW ID=v5.2.B Dell <platform> 060519 14:27:55 Version=0.0 ROM ID=0

Adr=0x46: FW ID=v5.2.B Dell <platform> 060519 14:27:55 Version=0.0 ROM ID=0

Adr=0x45: FW ID=v5.2.B Dell <platform> 060519 14:27:55 Version=0.0 ROM ID=0

.

.

.

Adr=0x41: FW ID=v5.2.B Dell <platform> 060519 14:27:55 Version=0.0 ROM ID=0

Adr=0x42: FW ID=v5.2.B Dell <platform> 060519 14:27:55 Version=0.0 ROM ID=0

```

Adr=0x43: FW ID=v5.2.B Dell <platform> 060519 14:27:55 Version=0.0 ROM ID=0
*** unit 0: ports capable of limited speed range cut-thru
0:bcmi_xgs5_bfd_init: uKernel BFD application not available
Current mode is now ESW
Diag NPU initialization over
Chip ID for PHY:0x0 is : 0x2399
EUROPA chip-ID LSB:2399 for PHY:60
EUROPA chip ID MSB and REV:0x80a0
Chip found!!! Initializing PHY....
Port mode:4 caui mode:1
Port mode:4 caui mode:1
_plp_europa_evora_pm4x25_pm_phys_enable::Checking core init variable:0
FW download Started
25% Downloaded
50% Downloaded
75% Downloaded
Firmware download success. GPREG2:0 retry:18
MST Dload Done:1
Slave Dload Done:3
FW version:0xd00b
_plp_europa_evora_pm4x25_pm_phys_enable::Checking core init variable:0
_plp_europa_evora_pm4x25_pm_port_init:: Initializing port
_plp_europa_evora_pm4x25_pm_port_init:: Initializing port
Core Ver:0
PHY ADDR: 0x60 device statistics counter increment enable settings:
    TCAM counters: Enabled
    Channel counters: Enabled
PHY ADDR: 0x60 device version major/minor/patch: 1/0/1
PHY ADDR: 0x60 device capabilities
    Egress only:          Yes
    Ingress only:         No
    Nof channels:         4
    Nof rules:            512
    Nof vPorts:           256

PHY ADDR: 0x60 device (EIP-217) version major/minor/patch: 1/0/3
PHY ADDR: 0x60 (EIP-217) capabilities
    Nof TCAM hit counters: 512
    TCAM hit counter size (bits): 64
    Nof TCAM packet counters: 1
    Nof TCAM byte counters: 0
    Nof TCAM global packet counters: 0
    Nof TCAM global byte counters: 0
    Nof TCAM channel packet counters: 7
    Nof TCAM channel byte counters: 0
    Nof channels:         4

bcm_plp_cfye_device_init API success for PHY-ID[96], macsec_side [0], return code [0]
PHY ADDR: 0x60 device statistics counter increment enable settings:
    SA counters: Enabled
    IFC counters: Enabled
    IFC1 counters: Enabled
    SecY counters: Enabled
    RxCAM counters: Enabled
    Channel counters: Enabled
PHY ADDR: 0x60 version major/minor/patch: 1/0/4
PHY ADDR: 0x60 capabilities
    Egress only:          Yes
    Ingress only:         No
    Nof SA's:             512
    Nof SC's:             256
    Nof channels:         4
    Nof vPorts:           256

PHY ADDR: 0x60 (EIP-64) version major/minor/patch: 1/0/2
PHY ADDR: 0x60 (EIP-64) capabilities
    AES present:          Yes
    AES feedback mode:    No
    FPGA solution:        No
    Galois fields s-boxes: No
    Lookup table s-boxes: Yes
    MACsec AES only:      Yes
    GHASH present:        Yes
    Once cycle core:      No

```

```

    AES speed:                15
    Key length:               3
    Parameter bits:          2
bcm_plp_secy_device_init API success for PHY-ID[96], macsec_side [0], return code [0]
PASSED: MACSec Initialize passed for PHY-ID[0x60], device-id [0]
PHY ADDR: 0x60 device statistics counter increment enable settings:
    TCAM counters: Enabled
    Channel counters: Enabled
PHY ADDR: 0x60 device version major/minor/patch: 1/0/1
PHY ADDR: 0x60 device capabilities
    Egress only:             No
    Ingress only:           Yes
    Nof channels:           4
    Nof rules:              512
    Nof vPorts:             256

PHY ADDR: 0x60 device (EIP-217) version major/minor/patch: 1/0/3
PHY ADDR: 0x60 (EIP-217) capabilities
    Nof TCAM hit counters:  512
    TCAM hit counter size (bits): 64
    Nof TCAM packet counters: 1
    Nof TCAM byte counters: 0
    Nof TCAM global packet counters: 0
    Nof TCAM global byte counters: 0
    Nof TCAM channel packet counters: 7
    Nof TCAM channel byte counters: 0
    Nof channels:          4

bcm_plp_cfye_device_init API success for PHY-ID[96], macsec_side [1], return code [0]
PHY ADDR: 0x60 device statistics counter increment enable settings:
    SA counters: Enabled
    IFC counters: Enabled
    IFC1 counters: Enabled
    SecY counters: Enabled
    RxCAM counters: Enabled
    Channel counters: Enabled
PHY ADDR: 0x60version major/minor/patch: 1/0/4
PHY ADDR: 0x60capabilities
    Egress only:             No
    Ingress only:           Yes
    Nof SA's:               512
    Nof SC's:               256
    Nof channels:           4
    Nof vPorts:             256

PHY ADDR: 0x60(EIP-64) version major/minor/patch: 1/0/2
PHY ADDR: 0x60(EIP-64) capabilities
    AES present:            Yes
    AES feedback mode:      No
    FPGA solution:         No
    Galois fields s-boxes: No
    Lookup table s-boxes:  Yes
    MACsec AES only:       Yes
    GHASH present:         Yes
    Once cycle core:       No
    AES speed:             15
    Key length:            3
    Parameter bits:        2

bcm_plp_secy_device_init API success for PHY-ID[96], macsec_side [1], return code [0]
PASSED: MACSec Initialize passed for PHY-ID[0x60], device-id [1]
config set speed = 25000
Port mode:0 caui mode:0
_plp_europa_evora_pm4x25_pm_phys_enable::Checking core init variable:0
_plp_europa_evora_pm4x25_pm_port_init:: Initializing port
Port mode:0 caui mode:0
_plp_europa_evora_pm4x25_pm_phys_enable::Checking core init variable:0
_plp_europa_evora_pm4x25_pm_port_init:: Initializing port
+++++
0: _bcm_esw_linkscan_update_port: Port xe12: Link Up with Speed 0M!
0: _bcm_esw_linkscan_update_port: Port xe13: Link Up with Speed 0M!
0: _bcm_esw_linkscan_update_port: Port xe14: Link Up with Speed 0M!
.
.
.

```


EEPROMTool

To program the type, length, value (TLV) format EEPROMs, use the `EEPROMTool`. You can also use the `EEPROMTool` to show all the TLV-formatted EEPROM contents or show specific EEPROM content by specifying the EEPROM type.

Tests

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

CLI options

```
root@dellemc-diag-os:~# EEPROMTool
DellEmc Diag - Eeprom Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

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
  Verify the EEPROM data:=
    EEPROMTool --eeprom=<eepromtype> --test --content (or)
    EEPROMTool -P <eepromtype> -t -c
  Verify EEPROM write protection:=
    EEPROMTool --eeprom=<eepromtype> --test --protect (or)
    EEPROMTool -P <eepromtype> -t -p
  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>...
  Modify FRU config file:=
    EEPROMTool --fru=<frutype> --modify=<filed> --value=<val> (or)
    EEPROMTool -T <frutype> -M <filed> -V <val>
  Add UUID to FRU config file:=
    EEPROMTool --fru=<frutype> --modify=<filed> --autogenuuid (or)
    EEPROMTool -T <frutype> -M <filed> -U
  Generate FRU data from the config file:=
    EEPROMTool --fru=<frutype> --file=<fru.bin> (or)
    EEPROMTool -T <frutype> -O <fru.bin>
  Write FRU data to EEPROM:=
    EEPROMTool --eeprom=<eepromtype> --fileprogram=<fru.bin> (or)
```

```

    eepromtool -P <eepromtype> -w <fru.bin>
Display EEPROM type:=
    eepromtool --eeprom=<eepromtype> --encoding (or)
    eepromtool -P <eepromtype> -E

Usage:=
common options:
-h, --h                Show the help text
-L, --listdevices     List all EEPROM devices
-P, --eeprom=         EEPROM type
-x, --show            Show operation
-I, --iteration=      Iteration command execution
-E, --encoding        EEPROM encoding type

specific options for TLV eeprom:
-l, --list            List operation
-e, --erase           Erase operation
-t, --test            Test using the pre-programmed configuration or use supplied config
-g, --get             Get operation
-s, --set             Set operation
-c, --content         EEPROM data validation Test option
-p, --protect         EEPROM write protection Test option

specific options for FRU eeprom:
-T, --fru             fru bin data operation
-M, --modify          fru bin modified tag
-V, --value           fru bin modified value
-O, --file            specify the output fru bin file
-w, --fileprogram     Write file to device
-U, --autogenuuid     Auto generate UUID and write to fru xml config file

specific options for mailbox eeprom:
-m, --psueepromdump  Dump the PSU EEPROM
-F, --faneepromdump  Dump the FAN EEPROM
root@dellemc-diag-os:~#

```

Output

```

root@dellemc-diag-os:~# 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
0xfc          EEPROM Test
0xfd          Vendor Extension
0xfe          CRC-32
root@dellemc-diag-os:~#
root@dellemc-diag-os:~# eepromtool --listdevices
IDEEPROM
FAN1EEPROM
FAN2EEPROM
FAN3EEPROM
PSU1EEPROM
PSU2EEPROM

root@dellemc-diag-os:~# eepromtool --eeprom=IDEEPROM --show
TlvInfo Header:

```

```

    Id String:    TlvInfo
    Version:      1
    Total Length: 185
TLV Name          Code Len Value
-----
Product Name      0x21  11 <platform>
Part Number       0x22   6 xxxxxx
Serial Number     0x23  20 TW0WYGRVDNT0091V0011
Base MAC Address  0x24   6 50:9A:4C:D8:D9:30
Manufacture Date  0x25  19 01/31/2019 15:49:20
Device Version    0x26   1 1
Label Revision    0x27   3 X00
Platform Name     0x28  32 x86_64-dellemc_<platform>_c3338-r0
MAC Addresses     0x2A   2 256
Manufacturer      0x2B   5 DNT00
Country Code     0x2C   2 TW
Vendor Name       0x2D   8 Dell EMC
Service Tag       0x2F   7 xxxxxx
Vendor Extension  0xFD   4 0x00 0x00 0x02 0xA2
Loader Version    0x29  10 x.xx.x.x-x
Diag Version      0x2E  11 x.xx.x.xx-x
CRC-32           0xFE   4 0xDB40BC2F
Checksum is valid.
root@dellemc-diag-os:~#

```

ethtool

The `ethtool` provides management interface details.

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

```

root@dellemc-diag-os:~# fantool
DellEmc Diag - Fan Controller Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03

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>]                    (or)
    fantool -t [-F <fan>]

Get the speed of the specified fan or all fans in RPM:=
    fantool --get --fan=<fan | all>                  (or)
    fantool -g -F <fan | all>

Set the fan(s) to the speed:=
    fantool --set --fan=<fan | all> --speed=<speed in RPM> (or)
    fantool -s -F <fan | all> -p <speed in RPM>

Run short test :=

```

```

    fantool --short-test                               (or)
    fantool -S
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>
Set the FPGA into normal mode:=
    fantool --reset-fpga-mode                           (or)
    fantool -Y -M

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, --reg=            Register
-V, --val=            Value to be set
-p, --speed=          Speed of the fan
-S, --short-test      Run the short test
-Y, --reset-fpga-mode Set FPGA to normal mode
  *Fans are from 1 to Max System Fans.
root@dellemc-diag-os:~#

```

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@dellemc-diag-os:~# fantool --test
Fan Controller Test .....
Setting speed for Fan 1 to 12000
Setting speed for Fan 2 to 12000
Setting speed for Fan 3 to 12000
Checking Fan Speeds
Fan 1 Speed: 18814 (12000)
Fan 1 Speed: 15855 (12000)
Fan 1 Speed: 12443 (12000)
Fan 2 Speed: 14672 (12000)
Fan 2 Speed: 11599 (12000)
Fan 3 Speed: 13195 (12000)
Setting speed for Fan 1 to 28000

```



```

Setting speed for Fan 2 to 28000
Setting speed for Fan 3 to 28000
Checking Fan Speeds
Fan 1 Speed: 15420 (28000)
Fan 1 Speed: 18460 (28000)
Fan 1 Speed: 21605 (28000)
Fan 1 Speed: 24887 (28000)
Fan 2 Speed: 24576 (28000)
Fan 3 Speed: 23831 (28000)
Setting speed for Fan 1 to 14000
Setting speed for Fan 2 to 14000
Setting speed for Fan 3 to 14000
Checking Fan Speeds
Fan 1 Speed: 22995 (14000)
Fan 1 Speed: 19960 (14000)
Fan 1 Speed: 16948 (14000)
Fan 1 Speed: 13993 (14000)
Fan 2 Speed: 14093 (14000)
Fan 3 Speed: 13559 (14000)
Setting speed for Fan 1 to 6000
Setting speed for Fan 2 to 6000
Setting speed for Fan 3 to 6000
Checking Fan Speeds
Fan 1 Speed: 11139 (6000)
Fan 1 Speed: 8141 (6000)
Fan 1 Speed: 6362 (6000)
Fan 2 Speed: 6105 (6000)
Fan 3 Speed: 6271 (6000)
Fan Controller Test: Overall test results----- >>>> PASSED
root@dellemc-diag-os:~#

root@dellemc-diag-os:~# fantool --get --fan=1
Fan 1 speed is 24730 RPM
root@dellemc-diag-os:~# fantool --get --fan=2
Fan 2 speed is 25700 RPM
root@dellemc-diag-os:~#

```

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 aligns 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

```

root@dellemc-diag-os:~# gpiotool
DellEmc Diag - GPIO Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

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 operation
-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@dellemc-diag-os:~# gpiotool --list
Chip 0 Core Gpio bits: 147 CORE gpiochip0
=====
Bit          Name      Dir  AC  Value
=====
 4          CPLD GPIO 4  OUT  LOW  0
 5              BOOT OK  OUT  LOW  0
 6          BOOT FINITO  OUT  LOW  0
 8          CPLD GPIO 8  OUT  LOW  0
 9          SYSTEM INT  IN   LOW  0
10          CPLD GPIO 10  OUT  LOW  0
22          CPLD GPIO 22  OUT  LOW  0
23          CPLD GPIO 23  OUT  LOW  0
37          CPLD GPIO 37  OUT  LOW  0
38          CPLD GPIO 38  OUT  LOW  0
39          1588 1PPS  IN   LOW  0
40          CPLD GPIO 40  OUT  LOW  0
100         CPLD GPIO 100  OUT  LOW  0
112         CPLD GPIO 112  OUT  LOW  0

```

get output

```

root@dellemc-diag-os:~# gpiotool --get --pin=4
Chip 0 Core Gpio bits: 147 CORE gpiochip0
=====
Bit          Name      Dir  Value
=====
 4          CPLD GPIO 4  OUT  0

```

set output

```

root@dellemc-diag-os:~# 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

```
root@dellemc-diag-os:~# i2ctool
DellEmc Diag - I2C Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

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_s
ize=<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> [--flag=<PEC>, ...]
    (or)
    i2ctool -w -b /dev/i2c-<bus_number> -a <address> -R <register> -W <width> -V <value> [-F
<PEC>, ...]
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>
Write without data:=
    i2ctool --no_data --bus=/dev/i2c-<bus_number> --
addr=<address>
    (or)
    i2ctool -N -b /dev/i2c-<bus_number> -a <address>

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
-N, --no_data          Write operation without data
-f, --config=<file_name> To specify the location of the config file e.g. /etc/dn/diag/
-C, --count=<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
-F, --flag=         Misc I2C flag, such as PEC write
-D, --display_size= Display size, {1,2,4} of bytesDisplay size, {1,2,4} of bytes
-I, --iteration=    Iteration command execution
root@dellemc-diag-os:~#

```

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@dellemc-diag-os:~# i2ctool
DellEmc Diag - I2C Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

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=<register> --
width=<width> --val=<value> [--flag=<PEC>, ...]                      (or)
    i2ctool -w -b /dev/i2c-<bus_number> -a <address> -R <register> -W <width> -V <value>
[-F <PEC>, ...]
  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>
  Write without data:=
    i2ctool --no_data --bus=/dev/i2c-<bus_number> --
addr=<address>
    (or)
    i2ctool -N -b /dev/i2c-<bus_number> -a <address>

```

```

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
-N, --no_data         Write operation without data
-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
-F, --flag=           Misc I2C flag, such as PEC write
-D, --display_size=  Display size, {1,2,4} of bytesDisplay size, {1,2,4} of bytes
-I, --iteration=      Iteration command execution
root@dellemc-diag-os:~#

```

scan and test Output

```

root@dellemc-diag-os:~# i2ctool --scan
/dev/i2c-0:
 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: 30 31 -- -- 34 35 36 -- -- -- -- -- -- --
40: -- -- -- -- 44 -- -- 48 -- -- -- -- -- --
50: 50 -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- RR RR RR RR RR RR RR
/dev/i2c-1:
 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: -- -- -- -- -- -- -- -- -- -- -- 2c -- -- --
30: -- 31 32 -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- 49 4a 4b 4c -- -- 4f
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- 71 -- -- -- -- -- -- RR RR RR RR RR RR RR
I2C devices found on bus #0: 8
 0x30 0x31 0x34 0x35 0x36 0x44 0x48 0x50
I2C devices found on bus #1: 9
 0x2c 0x31 0x32 0x49 0x4a 0x4b 0x4c 0x4f
 0x71

```

```

root@dellemc-diag-os:~# i2ctool --test
Testing I2C devices:

Checking I2C devices on bus 0:

+ Checking CPU board Thermal Sensor_TMP75AI 0x48 ..... Passed

+ Checking SPD EEPROM_M34E04 0x50 ..... Passed

Checking I2C devices on bus 1:

+ Checking CPU CPLD 0x31 ..... Passed

+ Checking SYSTEM CPLD 0x32 ..... Passed

+ Checking I2C Switch 0x71 ..... Passed

+ Checking Main Board ID 0x50 ..... Passed

```

```

+ Checking IR3595 1 0x70 ..... Passed
+ Checking IR3595 2 0x72 ..... Passed
+ Checking FAN Tray 01 EEPROM 0x50 ..... Passed
+ Checking FAN Tray 02 EEPROM 0x50 ..... Passed
+ Checking FAN Tray 03 EEPROM 0x50 ..... Passed
+ Checking PSU1 EEPROM 0x56 ..... Passed
+ Checking PSU1 FAN CONTROL 0x5e ..... Passed
+ Checking FAN Controller_EMCC2305 0x2c ..... Passed
+ Checking Thermal Sensor 1 0x49 ..... Passed
+ Checking Thermal Sensor 2 0x4a ..... Passed
+ Checking Thermal Sensor 3 0x4b ..... Passed
+ Checking Thermal Sensor 4 0x4c ..... Passed
+ Checking Thermal Sensor 5 0x4f ..... Passed
I2C Devices: Overall test results----- >>>> PASSED
root@dellemc-diag-os:~#

```

read Output

```

root@dellemc-diag-os:~# i2ctool --read --bus=/dev/i2c-0 --addr=0x50 --reg=0 --count=256
[00000000]: 0x12 0x11 0x0c 0x02 0x84 0x19 0x00 0x08 0x00 0x60 0x00 0x03 0x01 0x0b 0x80 0x00
|| .....`.....
[00000010]: 0x00 0x00 0x07 0x0d 0xf8 0x0f 0x00 0x00 0x6e 0x6e 0x6e 0x11 0x00 0x6e 0x20 0x08
|| .....nnn..n.
[00000020]: 0x00 0x05 0x70 0x03 0x00 0xa8 0x1b 0x28 0x28 0x00 0x78 0x00 0x14 0x3c 0x00 0x00
|| ..p....(.x.<..
[00000030]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x16 0x36 0x16 0x36
|| .....6.6
[00000040]: 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x00 0x00
|| .6.6.6.6.6.6..
[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 0x00 0x9c 0xb5 0x00 0x00 0x00 0x00 0xe7 0xd6 0x50 0xaf
|| .....P.
[00000080]: 0x11 0x11 0x23 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
|| ..#.....
[00000090]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
|| .....
[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 0x57
|| .....W
root@dellemc-diag-os:~#

```

write Output

```
/opt/dellemc/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@dellemc-diag-os:~# ledtool --test
LED Test Started... Will take few mins to complete.

Overall LED test result ====>> Passed
root@dellemc-diag-os:~#
```

CLI options

```
root@dellemc-diag-os:~# ledtool
DellEmc Diag - Led Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

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> [-S <instance>] [-T <state>] [-V <value>]
  Execute repeatedly command by count:=
    ledtool --iteration=max/<count>[option1] [option2]...
(or)
    ledtool -I max/<count> [option1] [option2]...
  Give control to CPU:=
    ledtool --set --cpu_ctrl
(or)
    ledtool -s -U
  Give control to FPGA:=
    ledtool --set --fpga_ctrl
(or)
    ledtool -s -Y
  Test using config file:=
    ledtool --test [--config=<config_file>]
(or)
    ledtool -t [-f <config_file>]
  Test LED manually
    ledtool --test --manual
(or)
    ledtool -t -m
  Set led pre and post settings :=
```

```

    ledtool --set --led=<led> --mode=<mode> [normal]
(or)
    ledtool -s -D <led> -M <mode>
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>
-U, --cpu_ctrl,       Set CPU control
-Y, --fpga_ctrl,      Set FPGA control
-M, --mode=,          Set CPLD Mode
-m, --manual,         Manual LED test

```

Output

list output

```

root@dellemc-diag-os:~# ledtool --list
Fan : options
      off solid-amber solid-green flashing-amber
System : options
      flashing-green solid-green solid-amber flashing-amber
Beacon : options
      off flashing-blue
Power : options
      off solid-amber solid-green flashing-amber
Stacking : options
      solid-green off
Stack-val : options
      0-F
7-digit : options
      off on flashing-green solid-green dot-off dot-on
root@dellemc-diag-os:~#

```

get Output

```

root@dellemc-diag-os:~# ledtool --get
Fan Led: off
System Led: flashing-green
Beacon Led: off
Power Led: solid-green
Stacking Led: off
Stack-valled: Value 0x38
7-digitled: on
root@dellemc-diag-os:~#

```

lpctool

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

The `lpctool` allows 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 DIAG-OS tools use `lpctool` to read LPC-connected registers.

CLI options

```
root@dellemc-diag-os:~# lpctool
DellEmc Diag - LPC Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

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

Usage:=
-f, --config=      To specify the location of the config file e.g. /etc/dn/diag/<file_name>
-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
-V, --val=        Value to be set
```

Output

Read output

```
root@dellemc-diag-os:~# lpctool --read --addr=0x102
Byte Port 0x102 :
[00000000]: 0xff  || .
```

Write output

```
root@dell-diag-os:/opt/dellemc/diag/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 writes can loop for several iterations, and works similar to the Address Read test.
- **Address Walking 1**—Walks a 1 through 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 completes 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 complete, 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 complete, the cell holds all the 1s.
- **Data Pattern**—Writes four 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

```
root@dellemc-diag-os:~# memtool
```

```
DellEmc Diag - Memory Tool  
version 2.0, x.xx.x.x-x  
build, 2019/07/03,
```

```
Syntax: memtool <option>
```

```
Show the Help-text:=  
    memtool --h  
    memtool -h  
Display the configuration info of the device:=  
    memtool --info  
    memtool -i  
List all of the memory regions in the config file:=  
    memtool --list  
    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>] [--be] (or)  
    memtool -r -a <address> -C <bytes> [-W <#8,16,32>] [-b]  
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 operation
-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 {8,16,32}
-b, --be            Accessed memory is in Big Endian Format
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 which requires two hyphens in front of the options. Options are required, optional, or none. If a parameter is required, 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 verbosity 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@dellemc-diag-os:~# memtool --list
=====
Region ID: 0
Region Name: DDR4
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 511
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@dellemc-diag-os:~#
root@dellemc-diag-os:~#
root@dellemc-diag-os:~# memtool --info
==== DDR4 SPD Data ====
Capacity 4096 MB, Density 4096 Mb, Rows: 15, Cols: 10
Bus Width: 64 bits, ECC: yes
Manufacturer: N/A, VendorID: 0
Part Number : N/A
Frequency : 2400 Mhz

[00000000]: 0x12 0x11 0x0c 0x02 0x84 0x19 0x00 0x08 0x00 0x60 0x00 0x03 0x01 0x0b 0x80 0x00
|| .....`.....
[00000010]: 0x00 0x00 0x07 0x0d 0xf8 0x0f 0x00 0x00 0x6e 0x6e 0x6e 0x11 0x00 0x6e 0x20 0x08
|| .....nnn..n .
[00000020]: 0x00 0x05 0x70 0x03 0x00 0xa8 0x1b 0x28 0x28 0x00 0x78 0x00 0x14 0x3c 0x00 0x00
|| ..p....(.x.<..
[00000030]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x16 0x36 0x16 0x36
|| .....6.6
[00000040]: 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x16 0x36 0x00 0x00
|| .6.6.6.6.6.6..
[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 0x00 0x9c 0xb5 0x00 0x00 0x00 0x00 0xe7 0xd6 0x50 0xaf
|| .....P.
[00000080]: 0x11 0x11 0x23 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
|| ..#.....
[00000090]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
|| .....
[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 0x57
|| .....W

```

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 : AW48M7228BNKOM
[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 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 0x00
|| .....
root@dell-diag-os:~#

```

Test output

```

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

```

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.

nputool

The `nputool` allows for configuring and testing the switch ASICs.

The `nputool` tests the NPU in the system. The `nputool` verifies that ports are up and traffic between the ports is working either using the CPU-generated packet or using IXIA connected to port-1 and port-2 based on the configuration.

Tests

Tests are shown in the following sections.

CLI options

The `nputool` shows the available options with the `nputool -h` or `nputool` command.

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

Syntax: nputool
  -h, --help           := Show this help
  -i, --init           := Initialize NPU chip
  -t, --test
        all            := Run All NPU tests
        id             := Run test based on test ID
  -s, --show
        counter        := Dump packet counters
        temp           := Display NPU temperature

  -l, --lpbk [phy/mac/ext]:= Specify Loopback type for traffic test
  -T, --traffic [ixia_self,ixia_adj,cpu_self,cpu_adj]
                        := Send IXIA or CPU traffic based on specified cfg
                        self->timbercon lpbk, adj->fiber lpbk
  -I, --iteration [count]:= Execute repeatedly command by count
  -v, --version        := Display version

Usage:
  nputool -i -t [all/0/1,2,3/4/./7] -T [ixia_self/ixia_adj/cpu_self/cpu_adj]
        -l [phy/mac/ext] := Run NPU tests based on user input

  nputool -I [count] -i -t 1 -T cpu_self := Run NPU test repeatedly by count
  nputool -i -s temp := Display NPU temperature
```

For the `nputool -i -t [all/0/1,2,3/4/./7]` usage:

- 0 = Test link status
- 1 = Test snake traffic
- 2 = Test prbs mac test
- 3 = Test prbs ext
- 4 = Test uplink link status
- 5 = Test uplink snake traffic
- 6 = Test uplink prbs mac
- 7 = Test uplink prbs ext

nputool version

```
root@dell-diag-os:/etc/dn/diag# nputool --v
Dell Diag nputool - version 1.0 sdk-6.5.3 package x.xx.x.x.xx 2016/08/12
root@dell-diag-os:/etc/dn/diag#

root@dell-diag-os:/etc/dn/diag# nputool --version
Dell Diag nputool - version 1.0 sdk-6.5.3 package x.xx.x.x.xx 2016/08/12
root@dell-diag-os:/etc/dn/diag#
```

Port link status test

- nputool -i -t 0
- nputool --i --test 0

```

root@dell-diag-os:~# root@dell-diag-os:/etc/dn/diag# nputool -i -t 0
8375_B0
-bash: root@dell-diag-os:/etc/dn/diag#: No such file or directory
sysconf_probe successful
global_sal_config successful
*** 1 BCM devices are detected
Diag NPU initialization over
    Test link_status_test for NPU 0 ..... Passed
    Test snake_traffic_test for NPU 0 ..... SKIPPED <<<---
        Test prbs_mac_test for NPU 0 ..... SKIPPED <<<---
        Test prbs_ext_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_link_status_test for NPU 0 ..... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_mac_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed

root@dell-diag-os:/etc/dn/diag# nputool -init -test 0
DMA pool size: 16777216
PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0, Driver BCM88375_B0
sysconf_probe successful
global_sal_config successful
*** 1 BCM devices are detected
Diag NPU initialization over
    Test link_status_test for NPU 0 ..... Passed
    Test snake_traffic_test for NPU 0 ..... SKIPPED <<<---
        Test prbs_mac_test for NPU 0 ..... SKIPPED <<<---
        Test prbs_ext_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_link_status_test for NPU 0 ..... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_mac_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed

```

CPU traffic with external loopback

Tests the traffic sent from the CPU internally generated packet to the front-end ports that are connected with external Loopback optics.

Connect all the ports with QSFP28 Loopback optics.

- nputool -i -t 1 -T cpu_self
- nputool --init --test 1 --traffic cpu_self

```

root@dell-diag-os:/opt/dellemc/diag/bin# nputool -i -t 1 -T cpu_self
DMA pool size: 16777216 PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0,
Driver BCM88375_B0 sysconf_probe successful global_sal_config successful ***
1 BCM devices are detected Diag NPU initialization over
Test link_status_test for NPU 0 ..... SKIPPED <<<---
Test snake_traffic_test for NPU 0 ..... Passed
Test prbs_mac_test for NPU 0 ..... SKIPPED <<<---
Test prbs_ext_test for NPU 0 ..... SKIPPED <<<---
Test uplink_link_status_test for NPU 0 .... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 .... SKIPPED <<<---
Test uplink_prbs_mac_test for NPU 0 .... SKIPPED <<<---
Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed

root@dell-diag-os:~# nputool --init --test 1 --traffic cpu_self
DMA pool size: 16777216 PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0,
Driver BCM88375_B0 sysconf_probe successful global_sal_config successful ***
1 BCM devices are detected Diag NPU initialization over
Test link_status_test for NPU 0 ..... SKIPPED <<<---
Test snake_traffic_test for NPU 0 ..... Passed
Test prbs_mac_test for NPU 0 ..... SKIPPED <<<---
Test prbs_ext_test for NPU 0 ..... SKIPPED <<<---
Test uplink_link_status_test for NPU 0 ..... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 .... SKIPPED <<<---
Test uplink_prbs_mac_test for NPU 0 ..... SKIPPED <<<---

```

```
Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed
```

CPU traffic with adjacent loopback

Tests the traffic sent from the CPU internally generated packet to the front-end ports which are connected with direct attach cables (DACs) or optics with cables connected top-to-bottom.

Connect all the ports with DACs or 40G/100G optics with cables.

- `nputool -i -t 1 -T cpu_adj`
- `nputool --init --test 1 --traffic cpu_adj`

```
root@dell-diag-os:~# nputool -i -t 1 -T cpu_adj
DMA pool size: 16777216
PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0, Driver BCM88375_B0
sysconf_probe successful
global_sal_config successful
*** 1 BCM devices are detected
Diag NPU initialization over
    Test link_status_test for NPU 0 ..... SKIPPED <<<---
    Test snake_traffic_test for NPU 0 ..... Passed
    Test prbs_mac_test for NPU 0 ..... SKIPPED <<<---
    Test prbs_ext_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_link_status_test for NPU 0 ..... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_mac_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed
root@dell-diag-os:~#
```

```
root@dell-diag-os:~# nputool -init -test 1 -traffic cpu_adj
DMA pool size: 16777216
PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0, Driver BCM88375_B0
sysconf_probe successful
global_sal_config successful
*** 1 BCM devices are detected
Diag NPU initialization over
    Test link_status_test for NPU 0 ..... SKIPPED <<<---
    Test snake_traffic_test for NPU 0 ..... Passed
    Test prbs_mac_test for NPU 0 ..... SKIPPED <<<---
    Test prbs_ext_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_link_status_test for NPU 0 ..... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_mac_test for NPU 0 ..... SKIPPED <<<---
    Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed
root@dell-diag-os:~#
```

IXIA traffic with external loopback

Tests the traffic sent from IXIA to port-1 and to the front-end ports which are connected with external Loopback optics.

Connect the first port to IXIA and all other ports with QSFP28 Loopback optics.

- `nputool -i -t 1 -T ixia_self -d`
- `nputool --init --test 1 --traffic ixia_self -d`

These commands configure the virtual local area network (VLAN) and after the BCM.0> shell displays, send the traffic from IXIA. To verify the counters, run the `show c` command in the BCM shell.

IXIA traffic with adjacent loopback

Tests the traffic sent from IXIA to the front-end ports which are connected with DACs or optics with cables connected top-to-bottom.

Connect the first two ports to IXIA and all the remaining ports with DACs or 40G/100G optics with cables.

- `nputool -i -t 1 -T ixia_adj`
- `nputool --init --test 1 --traffic ixia_adj`

The previous commands configure the VLAN and after the BCM.0> shell displays. To verify the counters, run the `show c` command in the BCM shell.

CPU traffic with external loopback for uplink ports (SFP+)

Traffic is sent from the CPU to the SFP+ ports.

Connect all the ports with the SFP+ optics with TX and RX shorted.

- `nputool -i -t 5 -T cpu_self`
- `nputool --init --test 5 --traffic cpu_self`

CPU traffic for uplink ports connected between adjacent ports

Traffic is sent from the CPU internally generated packet to the front-end Dell EMC SFP+ ports which are connected with SFP+ optics using a cable.

Connect the SFP+ ports with the Dell EMC SFP+ optics using cables.

- `nputool -i -t 5 -T cpu_adj`
- `nputool --i --test 5 --traffic cpu_adj`

IXIA traffic with external loopback

Traffic is sent from the CPU internally generated packet to the front-end SFP+ ports which are connected with the Dell EMC SFP+ optics using a cable.

Connect the first port to IXIA and all other ports with Loopback optics.

- `nputool -i -t 5 -T ixia_self -d`
- `nputool --init --test 5 --traffic ixia_self -d`

Configure the VLAN and display the BCM.0> shell. To verify the counters, use the `show c` command in the BCM shell.

IXIA traffic with adjacent ports connected to IXIA

Traffic is sent from the CPU internally generated packet to the front-end ports which are connected with DACs or optics using cables connected top-to-bottom.

Connect two ports to IXIA with SFP+ optics and cables.

- `nputool -i -t 5 -T ixia_adj`
- `nputool --init --test 5 --traffic ixia_adj`

Configure the VLAN and display the BCM.0> shell. To verify the counters, use the `show c` command in the BCM shell.

PRBS for QSFP ports

Connect ports with Loopback cables and run the PRBS MAC and EXT Loopback tests.

- PRBS MAC level test `nputool -i -t 2` or `nputool --init --test 2`
- PRBS EXT level test `nputool -i -t 3` or `nputool --init --test 3`

For example:

```
root@dell-diag-os:~# nputool --init --test 2
DMA pool size: 16777216 PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0,
  Driver BCM88375_B0 sysconf_probe successful global_sal_config successful ***
  1 BCM devices are detected Diag NPU initialization over
Test link_status_test for NPU 0 ..... SKIPPED <<<---
Test snake_traffic_test for NPU 0 ..... SKIPPED <<<---
Test prbs_mac_test for NPU 0 ..... Passed
Test prbs_ext_test for NPU 0 ..... SKIPPED <<<---
Test uplink_link_status_test for NPU 0 ..... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 ..... SKIPPED <<<---
Test uplink_prbs_mac_test for NPU 0 ..... SKIPPED <<<---
Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed

root@dell-diag-os:~# nputool --init --test 3
DMA pool size: 16777216 PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0,
  Driver BCM88375_B0 sysconf_probe successful global_sal_config successful ***
  1 BCM devices are detected Diag NPU initialization over
Test link_status_test for NPU 0 ..... SKIPPED <<<---
Test snake_traffic_test for NPU 0 ..... SKIPPED <<<---
Test prbs_mac_test for NPU 0 ..... SKIPPED <<<---
Test prbs_ext_test for NPU 0 ..... Passed
Test uplink_link_status_test for NPU 0 ..... SKIPPED <<<---
Test uplink_snake_traffic_test for NPU 0 ..... SKIPPED <<<---
```

```
Test uplink_prbs_mac_test for NPU 0 ..... SKIPPED <<<---
Test uplink_prbs_ext_test for NPU 0 ..... SKIPPED <<<---
NPU tests ..... Passed
```

PRBS for uplink ports

Connect the SFP+ ports with an external Loopback cable.

- PRBS MAC level test `./nputool -i -t 6`
- PRBS EXT level test `./nputool -i -t 7`

NPU temperature

Show the current NPU temperature.

- `nputool -i -s temp`
- `nputool --init --show temp`

```
root@dell-diag-os:~# nputool -i -s temp
DMA pool size: 16777216
PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0, Driver BCM88375_B0
sysconf_probe successful
global_sal_config successful
*** 1 BCM devices are detected
Diag NPU initialization over
NPU 0 Temperature
-----
monitor    current    peak
-----
0           41.8       44.2
1           41.3       44.7
2           36.4       38.8
3           39.8       43.2
-----
Average 39.8, maximum peak 44.7
```

```
root@dell-diag-os:~# nputool -init -show temp
DMA pool size: 16777216
PCI unit 0: Dev 0x8375, Rev 0x11, Chip BCM88375_B0, Driver BCM88375_B0
sysconf_probe successful
global_sal_config successful
*** 1 BCM devices are detected
Diag NPU initialization over
NPU 0 Temperature
-----
monitor    current    peak
-----
0           42.3       44.2
1           40.8       44.2
2           35.9       38.8
3           40.8       43.2
-----
Average 39.9, maximum peak 44.2
root@dell-diag-os:~#
```

Debugging

With traffic commands, use the `-d` option, which displays the `BCM.0>` shell. To check counters and if the link is up, use the `ps` and `show c` commands.

nvrantool

To read and write the NVRAM bits, use the `nvrantool`. The BIOS uses the NVRAM bits to control testing. The EDA tools also use the NVRAM bits.

The NVRAM is an area, usually in a battery backed-up device such as an RTC chip. The NVRAM bits do not change across reboots or power cycles. These bits control how devices boot and how the system performs tests. The `nvrantool` 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 register details it is writing. The `nvrantool` displays 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

```
root@dellemc-diag-os:~# nvrantool
DellEmc Diag - NVRAM Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

Syntax: nvrantool <option>
Show this help:=
nvrantool --h (or)
nvrantool -h
Read all or specific register NVRAM values:=
nvrantool --read [--reg=<register>] (or)
nvrantool -r [-R <register>]
Write NVRAM value:=
nvrantool --write [--reg=<register> --val=<value>] (or)
nvrantool -w [-R <register> -V <value>]
Execute repeatedly command by count:=
nvrantool --iteration=max/<count> [option1] [option2]... (or)
nvrantool -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@dellemc-diag-os:~# nvrantool --read
NVRAM Values:
0x00 0xdf 0xe6 0x03 0x03 0x00 0xea 0x3b

Test Status Fail Bits : offset 0x50 = 0x0
7 NVRAM test = 0
6 SSD test = 0
5 Reserved = 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 = 0xdf
7 NVRAM test = 1
6 SSD test = 1
5 Reserved = 0
4 PCI test = 1
3 Upper DRAM test = 1
2 Lower DRAM test = 1
1 ECC test = 1
0 SPD test = 1

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
```

```

7: 6 Reserved = 0
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
```

```
Fast POE Control Bits : offset 0x5b = 0x3b
```

```
0 Fast POE Enable = 1
```

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

Write output

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

opticstool

To check the presence or absence of optic devices, link status, and to read data from the optic devices' EEPROM, use the `opticstool`.

Tests

There are no tests on the optic devices. You can run a brief report that displays the optic presence or shows simple data, such as the serial number and device type. For more detailed information, use a device report.

CLI options

```

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

```

```
Syntax: opticstool <option>
```

```
Show the help-text:=
```

```

  opticstool --h
  opticstool -h

```

```
Show port and optics status:=
```

```

  opticstool --show[=brief] [--int=<interface>]
  opticstool -x[=brief] [-I <interface>]

```

```
Execute repeatedly command by count:=
```

```

  opticstool --iteration=max/<count> [option1] [option2]...
  opticstool -I max/<count> [option1] [option2]...

```

```
opticstool --read --int=<interface> [--page=<page #>] [--index=<offset>] [--cnt=<length>] (or)
```

```
opticstool -r -I <interface> [-p <page #>] [-i <offset>] [-C <length>]
```

```
opticstool --write --int=<interface> --page=<page #> --index=<offset> --val=<value> (or)
```

```
opticstool -w -i <interface> -p <page #> -i <offset> -V <value>
```

```
Usage:
```

```

-h, --h          Show the help text
-x, --show=     Show operation
-F, --int       Interface ID
-I, --iteration= Iteration command execution

```

```
-r, --read      Read operation
-w, --write     Write operation
```

- `show` —Shows information about the optic devices. With the `brief` option, only the ID and presence displays. Without the `brief` option, more details display, such as the serial number and device type. If you specify an interface, more detail displays about that device by reading the EEPROM.

Output

show=brief output

```
root@dell-diag-os:~# opticstool --show=brief
```

```
Show Optics in System (brief)
```

Port #	Name	Status
1	SFP+ 1	PRESENT
2	SFP+ 2	PRESENT
3	SFP+ 3	PRESENT
4	SFP+ 4	PRESENT
5	SFP+ 5	PRESENT
6	SFP+ 6	PRESENT
7	SFP+ 7	PRESENT
8	SFP+ 8	PRESENT
9	SFP+ 9	PRESENT
10	SFP+ 10	PRESENT
11	SFP+ 11	PRESENT
12	SFP+ 12	PRESENT
13	SFP+ 13	PRESENT
14	SFP+ 14	PRESENT
15	SFP+ 15	PRESENT
16	SFP+ 16	PRESENT
17	SFP+ 17	PRESENT
18	SFP+ 18	PRESENT
19	SFP+ 19	PRESENT
20	SFP+ 20	PRESENT
21	SFP+ 21	PRESENT
22	SFP+ 22	PRESENT
23	SFP+ 23	PRESENT
24	SFP+ 24	PRESENT
25	SFP+ 25	PRESENT
26	SFP+ 26	PRESENT
27	SFP+ 27	PRESENT
28	SFP+ 28	PRESENT
29	SFP+ 29	PRESENT
30	SFP+ 30	PRESENT
31	SFP+ 31	PRESENT
32	SFP+ 32	PRESENT
33	SFP+ 33	PRESENT
34	SFP+ 34	PRESENT
35	SFP+ 35	PRESENT
36	SFP+ 36	PRESENT
37	SFP+ 37	PRESENT
38	SFP+ 38	PRESENT
39	SFP+ 39	PRESENT
40	SFP+ 40	PRESENT
41	QSFP+ 41	PRESENT
42	QSFP+ 42	PRESENT
43	QSFP28 43	PRESENT
44	QSFP28 44	PRESENT
45	QSFP28 45	PRESENT
46	QSFP28 46	PRESENT
47	QSFP28 47	PRESENT
48	QSFP28 48	PRESENT

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

show output

```

root@dell-diag-os:~# opticstool --show
Show Optics in System
Port # Name Status Type Part Number Rev Serial Number
-----
1 SFP+ 1 PRESENT SFP 616740000 B CNOC6Y7M41A0
2 SFP+ 2 PRESENT SFP 616740000 B CNOC6Y7M41A0
3 SFP+ 3 PRESENT SFP 616740000 C CNOC6Y7M01I4
4 SFP+ 4 PRESENT SFP 616740000 C CNOC6Y7M01I4
5 SFP+ 5 PRESENT SFP 616740000 C CNOC6Y7M490B@
6 SFP+ 6 PRESENT SFP 616740000 C CNOC6Y7M490B@
7 SFP+ 7 PRESENT SFP 616740000 C CNOC6Y7M490BDD
8 SFP+ 8 PRESENT SFP 616740000 C CNOC6Y7M490BDD
9 SFP+ 9 PRESENT SFP 616740000 C CNOC6Y7M482HV@
10 SFP+ 10 PRESENT SFP 616740000 C CNOC6Y7M482HV@
11 SFP+ 11 PRESENT SFP 616740000 C CNOC6Y7M490BEL
12 SFP+ 12 PRESENT SFP 616740000 C CNOC6Y7M490BEL
13 SFP+ 13 PRESENT SFP 616740000 C CNOC6Y7M490BD
14 SFP+ 14 PRESENT SFP 616740000 C CNOC6Y7M490BD
15 SFP+ 15 PRESENT SFP 616740000 C CNOC6Y7M490BDD
16 SFP+ 16 PRESENT SFP 616740000 C CNOC6Y7M490BDD
17 SFP+ 17 PRESENT SFP 616740000 C CNOC6Y7M48A2E@
18 SFP+ 18 PRESENT SFP 616740000 C CNOC6Y7M48A2E@
19 SFP+ 19 PRESENT SFP 616740000 C CNOC6Y7M482@@@
20 SFP+ 20 PRESENT SFP 616740000 C CNOC6Y7M482@@@
21 SFP+ 21 PRESENT SFP 616740000 C CNOC6Y7M48C2MP@
22 SFP+ 22 PRESENT SFP 616740000 C CNOC6Y7M48C2MP@
23 SFP+ 23 PRESENT SFP 616740000 C CNOC6Y7M40A0HB
24 SFP+ 24 PRESENT SFP 616740000 C CNOC6Y7M40A0HB
25 SFP+ 25 PRESENT SFP 616740000 C CNOC6Y7M41A0BP
26 SFP+ 26 PRESENT SFP 616740000 C CNOC6Y7M41A0BP
27 SFP+ 27 PRESENT SFP 616740000 C CNOC6Y7M411J
28 SFP+ 28 PRESENT SFP 616740000 C CNOC6Y7M411J
29 SFP+ 29 PRESENT SFP 616740000 C CNOC6Y7M41A0BR
30 SFP+ 30 PRESENT SFP 616740000 C CNOC6Y7M41A0BR
31 SFP+ 31 PRESENT SFP 616740000 C CNOC6Y7M40A0HB
32 SFP+ 32 PRESENT SFP 616740000 C CNOC6Y7M40A0HB
33 SFP+ 33 PRESENT SFP 616740000 C CNOC6Y7M49M4BG5
34 SFP+ 34 PRESENT SFP 616740000 C CNOC6Y7M49M4BG5
35 SFP+ 35 PRESENT SFP 616740000 C CNOC6Y7M49M4BEJ
36 SFP+ 36 PRESENT SFP 616740000 C CNOC6Y7M49M4BEJ
37 SFP+ 37 PRESENT SFP 599700001 A APF11370018C9V
38 SFP+ 38 PRESENT SFP 599700001 A APF11370018C9V
39 SFP+ 39 PRESENT SFP 616740000 C CNOC6Y7M48C2MUP
40 SFP+ 40 PRESENT SFP 616740000 C CNOC6Y7M48C2MUP
41 QSFP+ 41 PRESENT QSFP+ 599690001 D APF11510011VRR
42 QSFP+ 42 PRESENT QSFP+ AFBR-79E4Z-D-FT1 01 QB382231
43 QSFP28 43 PRESENT QSFP28 1002971101 1 504020274
44 QSFP28 44 PRESENT QSFP28 1002971101 1 504020274
45 QSFP28 45 PRESENT QSFP28 1002971051 1 506220006
46 QSFP28 46 PRESENT QSFP28 1002971051 1 506220006
47 QSFP28 47 PRESENT QSFP28 1002971101 1 504120586
48 QSFP28 48 PRESENT QSFP28 1002971101 1 504120586
root@dell-diag-os:~#

```

show --int=interface # output

```

root@dell-diag-os:~# opticstool --show --int=48
Show Optics in System
=====
QSFP28 48 Detailed Display
=====
Link Status
-----
Port Status
Loss of Signal      :
RX Signal Lock Error :
PCS Link State     :

```

```

Link Faults      :
Remote          :
Local           :
Idle Error      :
Illegal Symbol  :
Error Symbol    :
Present         : Present
Device Data:
[00000000]: 0x11 0x05 0x06 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || .....
[00000010]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || .....
[00000020]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || .....
[00000030]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 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 0x38 0x00
  || .....8.
[00000070]: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || .....
[00000080]: 0x11 0x00 0x23 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
  || ..#.
[00000090]: 0x00 0x00 0x01 0xa0 0x4d 0x6f 0x6c 0x65 0x78 0x20 0x49 0x6e 0x63 0x2e 0x20 0x20
  || ...Molex Inc.
[000000a0]: 0x20 0x20 0x20 0x20 0x00 0x00 0x09 0x3a 0x31 0x30 0x30 0x32 0x39 0x37 0x31 0x31
  || ...:10029711
[000000b0]: 0x30 0x31 0x20 0x20 0x20 0x20 0x20 0x20 0x31 0x20 0x00 0x00 0x00 0x00 0x00 0x4c
  || 01      1 .....L
[000000c0]: 0x00 0x00 0x00 0x00 0x35 0x30 0x34 0x31 0x32 0x30 0x35 0x38 0x36 0x20 0x20 0x20
  || ...504120586
[000000d0]: 0x20 0x20 0x20 0x20 0x31 0x35 0x30 0x32 0x31 0x30 0x20 0x20 0x00 0x00 0x00 0x18
  ||      150210 ....
[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 0x00
  || .....
Vendor:      Molex Inc.
Part No:    1002971101
Revision:   1
Serial Num: 504120586
ID          : 0x11
Extended ID : 0x00
Connector   : 0x23
Specification : 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Encoding    : 0x00
BR Nominal  : 0x00
Length (9um) Km      : 0x00
Length (9um) 100m   : 0x00
Length (50um) 10m   : 0x00
Length (62.5um) 10m : 0x00
Length (copper) 10m : 0x01
Cable Attenuation   : 0x00 (2.5 Ghz) 0x00 (5.0 Ghz)
CheckCodeBase      : 0x4c (0x4c)
-----
Extended ID Fields
-----
Options          :
BR Max           :
BR Min           :
Date Code        : 2015-02-10
CheckCodeExt     : 0x18 (0x18)
TX Output Disable : Not Disabled
-----
Diagnostics Information
-----
Module Monitoring Values:
Current Temp:    0.000 (Celsius)
Supply Volts:   0.000 (Volts)
Channel Monitoring Values:

```

```
Recv:      0.000      0.000      0.000      0.000 (dBm)
Bias:      0.000      0.000      0.000      0.000 (mA)
root@dell-diag-os:~#
```

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 endianness.

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 config 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 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 0xffffffff0.
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 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 that 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 is displayed.

CLI options

```

root@dellemc-diag-os:~# pltool
DellEmc Diag - Programable Logic Tool
version x.x, x.xx.x.x-x
build, yyyy/mm/dd,

Syntax: pltool <option>
  Show this help text:=
    pltool --h                               (or)
    pltool -h
  Test (RW) the scratchpad registers:=
    pltool --test                             (or)
    pltool -t
  Test default & writable registers:=
    pltool --test --default                   (or)
  Test Port interrupt:=
    pltool --test --interrupt                 (or)
    pltool -t -i
  Test Reset:=
    pltool --test --reset                     (or)
    pltool -t -s
  Test Low Power mode:=
    pltool --test --lowpower                  (or)
    pltool -t -p
  List devices and registers:=
    pltool --list [--lstype=<devicetype>]     (or)
    pl 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 their current 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
-F, --default         Test option to test default registers
-i, --interrupt       Test option to test Interrupt
-s, --reset           Reset Test option
-p, --lowpower        Low Power Test option
-l, --list            List operation
-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 must 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
```

poetool

To use the poetool, connect ports 1/13/25/37 to the device.

 **NOTE:** The N2200-ON Series and N3200-ON Series platforms do not support this tool.

Tests

To enable POE+ power before testing, use the following poetool command:

```
Command: i2ctool -w -b /dev/i2c-0 -a 0x28 -R 0xe -V 0xff
```

- Command 1: `poetool -i` This command is initial poetool.
- Command 2: `poetool -s -p 0 -m af_mode` This sets the af mode.
- Command 3: `poetool -g -P` This is the read value.

```
DIAG:/ # poetool -v
DellEmc Diag poetool - version x.x package x.xx.x.x-x 2018/02/08
DIAG:/ # i2ctool -w -b /dev/i2c-0 -a 0x28 -R 0xe -V 0xff
DIAG:/ # i2ctool -r -b /dev/i2c-0 -a 0x28 -R 0xe
[0000000e]: 0xff || .
DIAG:/ # poetool -i
DIAG:/ # poetool -s -p 0 -m disable
```

```
DIAG:/ # poetool -g -P
```

Port	Power
1	13.5
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0
10	0.0
11	0.0
12	0.0
13	13.5
14	0.0
15	0.0
16	0.0
17	0.0
18	0.0
19	0.0
20	0.0
21	0.0
22	0.0
23	0.0
24	0.0
25	13.5
26	0.0
27	0.0
28	0.0
29	0.0
30	0.0
31	0.0
32	0.0
33	0.0
34	0.0
35	0.0
36	0.0
37	13.5
38	0.0
39	0.0
40	0.0
41	0.0
42	0.0
43	0.0
44	0.0
45	0.0
46	0.0
47	0.0
48	0.0

```
DIAG:/ # poetool -g -B
```

```
Power Budget:          1612
Max shutdown voltage:  57.0
Min shutdown voltage:  44.0
Guard Band:           1
Source Type:          0
```

```
DIAG:/ # poetool -g -p 1 -T
```

```
Port 1  Status:      0x01
DIAG:/ # poetool -g -p 2 -T
```

```
Port 2  Status:      0x1b
DIAG:/ # poetool -s -p 0 -F enable
DIAG:/ # poetool -g -p 1 -T
```

```
Port 1  Status:      0x02
DIAG:/ # poetool -s -p 0 -F disable
DIAG:/ # poetool -g -p 1 -T
```

```
Port 1  Status:      0x01
DIAG:/ # poetool -g -S
```



```

HW Version:          0x0
Product Number:     14
SW Vversion:        0265
Parameter Number:   0098
Build Number:       0x1
Internal SW version: 639
DIAG:/ # poetool -g -D

```

```

Device 0
Version:           129
ASIC status:       0x1
IC-HW:             0x8
Temp:              51 C
TSH                110

```

```

Device 1
Version:           129
ASIC status:       0x1
IC-HW:             0x8
Temp:              47 C
TSH                110

```

```

Device 2
Version:           129
ASIC status:       0x1
IC-HW:             0x8
Temp:              47 C
TSH                110

```

```

.
.
.
.

```

CLI options

```

Syntax: poetool <option>
Show the help-text:=
    poetool --help                (or)
    poetool -h
Display version:=
    poetool --version            (or)
    poetool -v
Reset the PoE controller:=
    poetool --reset [--config=<config_file>] (or)
    poetool -R [-f <config_file>]
Restore default setting of PoE controller:=
    poetool --restore [--config=<config_file>] (or)
    poetool -r [-f <config_file>]
Init matrix of PoE controller:=
    poetool --init [--config=<config_file>] (or)
    poetool -i [-f <config_file>]
Get all PoE port power:=
    poetool --get --port-power [--config=<config_file>] (or)
    poetool -g -P [-f <config_file>]
Get the power budget of PoE controller:=
    poetool --get --budget [--config=<config_file>] (or)
    poetool -g -B [-f <config_file>]
Get the new port status of PoE controller:=
    poetool --get --port=<port> --port-status [--config=<config_file>]
(or)
    poetool -g -p <port> -T [-f <config_file>]
Get the software version of PoE controller:=
    poetool --get --swversion [--config=<config_file>]
(or)
    poetool -g -S [-f <config_file>]
Get the device status of PoE controller:=
    poetool --get --device-status [--config=<config_file>]
(or)
    poetool -g -D [-f <config_file>]

```

```

Get the port measurement of PoE controller:=
poetool --get --port=<port | 0 for all port> --measurement [--config=<config_file>]
(or)
poetool -g -p <port | 0 for all port> -M [-f <config_file>]
Get the value of Individual Mask Key Number:=
poetool --get --mask <key number>
(or)
poetool -g -K <key number>
Set the power budget of PoE controller:=
poetool --set --budget=<budget> [--config=<config_file>]
(or)
poetool -s -B<budget> [-f <config_file>]
Set the PoE legacy mode:=
poetool --set --legacy=<enable | disable> [--config=<config_file>]
(or)
poetool -s -l <enable | disable> [-f <config_file>]
Set the PoE 4pair port mode:=
poetool --set --port=<port | 0 for all port> --4pair-mode=<enable | disable> [--
config=<config_file>] (or)
poetool -s -p <port | 0 for all port> -F <enable | disable> [-f <config_file>]
Set the PoE port mode:=
poetool --set --port=<port | 0 for all port> --mode=<af_mode | at_mode | poh_mode |
disable> [--config=<config_file>] (or)
poetool -s -p <port | 0 for all port> -m <af_mode | at_mode | poh_mode | disable> [-f
<config_file>]
Set the value of Individual Mask Key Number:=
poetool --set --mask <key number> --mask-value <enable |
disable> (or)
poetool -s -K <key number> -V <enable | disable>
Usage:=
-h, --help          Show the help text
-v, --version       Display version
-f, --config=<file_name> To specify the location of the config file e.g. /etc/dn/diag/
-R, --reset         Reset controller
-r, --restore       Restore default setting
-i, --init          Initilize POE controller
-p, --port=        Port Id
-g, --get           Get operation
-s, --set           Set operation
-m, --mode=        Mode of port
-B, --budget=      Power Budget
-P, --port-power   Power of port
-l, --legacy=      Legacy mode
-S, --swversion    Software version
-D, --device-status PoE device status
-T, --port-status  New port status
-M, --measurement  Port measurement
-F, --4pair-mode=  4Pair port mode
-K, --mask=        Individual Mask Key Number
-V, --mask-value=  Value of Mask

```

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

```
root@dellemc-diag-os:~# psutool
DellEmc Diag - Power Supply Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

Syntax: psutool <option>
Show the Help-text:=
    psutool --h
    psutool -h
    (or)
Test using the default config file:=
    psutool --test [--supply=<power_supply>]
    psutool -t [-S <power_supply>]
    (or)
Set the Fan Speed on the Power Supply:=
    psutool --fanspeed=<percentage> [--supply=<power_supply>]
    psutool -F <percentage> [-S <power_supply>]
    (or)
Read the register on the Power Supply:=
    psutool --read --supply=<power_supply> --reg=<register>
    psutool -r -S <power_supply> -r <register>
    (or)
Write the value into the Power Supply Register:=
    psutool --write --supply=<power_supply> --reg=<register> --val=<value>
    psutool -w <power_supply> -R <register> -V <value>
    (or)
Execute repeatedly command by count:=
    psutool --iteration=max/<count> [option1] [option2]...
    psutool -I max/<count> [option1] [Usage:=
    -h, --h          Show the help text
    -t, --test      Test using the pre-programmed configuration or use supplied config
    -F, --fanspeed  Fan Speed
    -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
```

test option

```
root@dellemc-diag-os:~# psutool --test

Power Supply Test all
DELL PSU FAN Speed 4512 RPM
DELL PSU Current IN 0.8(A)
DELL PSU Current OUT 2.8(A)
DELL PSU Voltage IN 206.0(V)
DELL PSU Voltage Out 56.4(V)
DELL PSU Power IN 154.8(W)
DELL PSU Power Out 155.2(W)
DELL PSU Temperature 31.2(C)
PSU-1 ..... Passed
PSU-2 is not present
Power Supply Test: Overall test results----- >>>> PASSED
root@dellemc-diag-os:~#
```

rtctool

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

Tests

CLI options

```
root@dellemc-diag-os:~# rtctool
DellEmc Diag - RTC Tool
version 1.1, x.xx.x.x-x
build, 2019/07/03,

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

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

Test options

```
root@dellemc-diag-os:~# rtctool --test
Testing RTC Devices .....
Testing RTC Device for rollover .....
Set Current RTC date to 1/1/2000, RTC time to 00:00:59.
```

```

Set Current RTC date to 1/1/2000, RTC time to 00:59:59.
Set Current RTC date to 1/1/2000, RTC time to 23:59:59.
Set Current RTC date to 1/31/2000, RTC time to 23:59:59.
Set Current RTC date to 12/31/2000, RTC time to 23:59:59.
Set Current RTC date to 7/23/2019, RTC time to 18:07:56.
RTC Devices: Overall test results----- >>>> PASSED
root@dellemc-diag-os:~#

```

smbiostool

The `smbiostool` displays information about the BIOS and also reprograms the BIOS flash.

NOTE: The `smbiostool` is not available for all platforms. For some platforms, this tool is replaced with the `updatetool`.

CLI options

```

root@dellemc-diag-os:~# smbiostool
DellEmc Diag - SMBIOS Tool
version 1.3, x.xx.x.x-x
build, 2019/07/03,

Usage:=
    smbiostool --h                               (or)
    smbiostool -h
Test whether the right SPI flash detected for the platform:=
    smbiostool --test                             (or)
    smbiostool -t
Print the BIOS version:=
    smbiostool --biosversion                       (or)
    smbiostool -b
Check whether the SPI flash was detected:=
    smbiostool --biosflashdetect                   (or)
    smbiostool -f
Take a backup of the Current running BIOS:=
    smbiostool --biosread <FILEPATH>               (or)
    smbiostool -r
Update the bios:=
    smbiostool --biosupdate <FILEPATH>             (or)
    smbiostool -u <FILEPATH>
Take a backup of the Current BIOS NVRAM region:=
    smbiostool --getbiosnvram <FILEPATH>           (or)
    smbiostool -g <FILEPATH>
Update the bios with the specified BIOS NVRAM:=
    smbiostool --setbiosnvram <FILEPATH>           (or)
    smbiostool -s <FILEPATH>
Dump the DMI table:=
    smbiostool --biosdumpall                       (or)
    smbiostool -d
Check whether the SPI flash was detected:=
    smbiostool --biosdumpfields <*options*>        (or)
    smbiostool -S <*options*>
<*options*> for biosdumpfields:=
    -q          Less verbose output
    -s          Only display the value of the given DMI string
    -t          TYPE Only display the entries of given type
    -u          Do not decode the entries
    --dump-bin <FILE> Dump the DMI data to a binary file
    --from-dump <FILE> Read the DMI data from a binary file
    -V          Display the version of dmidecode binary
root@dellemc-diag-os:~#

```

Output

```
root@dellemc-diag-os:~# smbiostool --biosversion
x.xx.x.x-x
root@dellemc-diag-os:~#
root@dellemc-diag-os:~# smbiostool --biosflashdetect
Warning: BIOS region SMM protection is enabled!
Warning: Setting Bios Control at 0xdc from 0xaa to 0x89 failed.
New value is 0xaa.
Warning: SPI Configuration Lockdown activated.
FREG2: Warning: Management Engine region (0x00001000-0x01595fff) is locked.
Not all flash regions are freely accessible by flashrom. This is most likely
due to an active ME. Please see http://flashrom.org/ME for details.
PR0: Warning: 0x01860000-0x01e6efff is read-only.
PR1: Warning: 0x01e70000-0x01ffffff is read-only.
Found Macronix flash chip "MX25L25635F" (32768 kB, Programmer-specific) at physical address
0x0.
root@dellemc-diag-os:~#
```

storagetool

The storagetool tests mounted storage media.

The tool searches for any device in `/dev/hd*`, `sda`, `sdb`, or `sd` 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

```
root@dellemc-diag-os:~# storagetool
DellEmc Diag - Storage Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

Syntax: storagetool <option>
  Show the help-text:=
    storagetool --h                               (or)
    storagetool -h
  Mount usb device when inserted (mandatory):=
    storagetool --mountusb [--dev=<device>]        (or)
    storagetool -m [-D <device>]
  Unmount usb device before removed (mandatory):=
    storagetool --unmountusb [--dev=<device>]      (or)
    storagetool -u [-D <device>]
  List devices:=
    storagetool --list                             (or)
    storagetool -l
  Test devices(empty for all):=
    storagetool --test [--dev=<device>]            (or)
    storagetool -t [-D <device>]
  Test devices(empty for all) with bootorder:=
    storagetool --test [--bootorder]              (or)
    storagetool -t [-b]
  Get the smart status for a device
    storagetool --smart --dev=<device>             (or)
    storagetool -S -D <device>
  Get the emmc parameters for a device
    storagetool --mmc --dev=<device>               (or)
    storagetool -M -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
    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
-M, --mmc             Print eMMC parameters
-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
root@dellemc-diag-os:~#

```

Output

list output

```

root@dellemc-diag-os:~# storagetool --list
Mounted Filesystem Devices:
/dev/mmcblk0p2 / ext4
root@dellemc-diag-os:~#

```

test output

```

root@dellemc-diag-os:~# storagetool --test --dev=/dev/mmcblk0
Testing Storage Devices ..... Passed
Testing UEFI mode
Checking UEFI sysfs ..... Passed
Checking EFI System Partition Presence ..... Passed
root@dellemc-diag-os:~#

```

smart output

```

root@dellemc-diag-os:~# storagetool --smart -dev=/dev/sda2
smartctl x.x 2013-07-26 r3841 [x86_64-linux-4.9.30] (local build)
Copyright (C) 2002-13, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Device Model:          M.2 (S42) 3IE3
Serial Number:         BCA11808150490067
LU WWN Device Id:     5 24693f 2ca223959
Firmware Version:     S17411i
User Capacity:         8,012,390,400 bytes [8.01 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:       ATA8-ACS (minor revision not indicated)
SATA Version is:      SATA 3.0, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is:        Tue Jul 23 18:12:07 2019 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.
No General Purpose Logging support.
```

```
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	0x0013	100	100	001	Pre-fail	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	202	000	000	Old_age	Always	- 458
10	Unknown_SSD_Attribute	0x0000	000	000	000	Old_age	Offline	- 0
12	Power_Cycle_Count	0x0002	079	000	000	Old_age	Always	- 79
163	Unknown_Attribute	0x0000	000	000	000	Old_age	Offline	- 9
168	Unknown_Attribute	0x0000	000	000	000	Old_age	Offline	- 0
169	Unknown_Attribute	0x0000	099	000	000	Old_age	Offline	- 99
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
194	Temperature_Celsius	0x0000	030	100	000	Old_age	Offline	- 30
	(Min/Max 2/100)							
197	Current_Pending_Sector	0x0012	000	100	000	Old_age	Always	- 0
225	Unknown_SSD_Attribute	0x0000	000	065	000	Old_age	Offline	- 148924
240	Unknown_SSD_Attribute	0x0000	000	000	000	Old_age	Offline	- 0
165	Unknown_Attribute	0x0002	010	001	000	Old_age	Always	- 10
167	Unknown_Attribute	0x0002	006	001	000	Old_age	Always	- 6
170	Unknown_Attribute	0x0003	100	001	000	Pre-fail	Always	- 163
171	Unknown_Attribute	0x0002	000	001	000	Old_age	Always	- 0
172	Unknown_Attribute	0x0002	000	001	000	Old_age	Always	- 0
176	Erase_Fail_Count_Chip	0x0000	100	001	000	Old_age	Offline	- 0
187	Reported_Uncorrect	0x0002	000	001	000	Old_age	Always	- 0
229	Unknown_Attribute	0x0002	100	001	000	Old_age	Always	-
	89569723955864							
232	Available_Reservd_Space	0x0003	100	001	000	Pre-fail	Always	- 0
235	Unknown_Attribute	0x0002	000	000	000	Old_age	Always	- 0
241	Total_LBAs_Written	0x0002	100	001	000	Old_age	Always	- 581
242	Total_LBAs_Read	0x0002	100	001	000	Old_age	Always	- 585

Read SMART Error Log failed: scsi error badly formed scsi parameters

Read SMART Self-test Log failed: scsi error badly formed scsi parameters

Selective Self-tests/Logging not supported

root@dellemc-diag-os:~#

bonnie output

```
root@dellemc-diag-os:~# storagetool --bonnie -dev=/dev/sda2
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
dellemc-diag-o 250M 31172 99 443651 99 +++++ +++ 27247 99 +++++ +++ +++++ +++
-----Sequential Create----- -----Random Create-----
-Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
32 31268 72 +++++ +++ 58250 98 26991 99 +++++ +++ 41123 99
dellemc-diag-os,250M,31172,99,443651,99,+++++,+++,27247,99,+++++,+++,+++++,+++,32,31268,72,+++
++,+++,58250,98,26991,99,+++++,+++,41123,99
root@dellemc-diag-os:~#

```

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.

```

root@dellemc-diag-os:~# temptool
DellEmc Diag - Temperature Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

Syntax: temptool <option>
  Show the help-text:=
    temptool --h
    temptool -h
  Test the pre-programmed configuration:=
    temptool --test --config=<config_file> (or)
    temptool -t -f <config_file>
  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> (or)
    temptool -x -f <config_file>

  Get the reg of a temp sensor:=
    temptool --read --sensor=<sensor tag> --reg=<ofs> --width=<reg size 1B/2B/3B> --
count=<Byte count> (or)
    temptool -r -D <tag> -R <ofs> -W <width> -C <byte count>
  Set the reg of a temp sensor:=
    temptool --write --sensor=<sensor tag> --reg=<ofs> --width=<reg size 1B/2B/3B> --

```

```

count=<Byte count> --val=<val1,val2..> (or)
      temptool -w -D <tag> -R <ofs> -W <width> -C <byte count> -V <values>
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
  -w, --write           Write operation
  -r, --read           Read operation
  -W, --width          Width {8,16}
  -D, --sensor
  -V, --val            Count
  -C, --count
root@dellemc-diag-os:~#

```

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

Output

test output

```

root@dellemc-diag-os:~# temptool --test
Testing Temp sensor devices:
+ Checking [CPU board Thermal Sensor_TMP75AI] = 30.0 C ..... Passed
+ Checking [Temp Sensor 1] = 29.0 C ..... Passed
+ Checking [Temp Sensor 2] = 41.0 C ..... Passed
+ Checking [Temp Sensor 3] = 27.0 C ..... Passed
+ Checking [Temp Sensor 4] = 30.0 C ..... Passed
+ Checking [Temp Sensor 5] = 28.0 C ..... Passed
Temp Sensors: Overall test results----- >>>> PASSED
root@dellemc-diag-os:~#

```

updatetool

The updatetool shows the CPLD, FPGA, and BIOS versions that are used to upgrade the CPLD, FPGA, and BIOS.

Tests

There are no defined tests with updatetool.

CLI options

```

root@dellemc-diag-os:~# updatetool
DellEmc Diag - Update Tool
version 2.0, x.xx.x.x-x
build, 2019/07/03,

Syntax: updatetool <option>
Print the Help-Text:=
      updatetool --help                (or)
      updatetool -h

Tool Version:=
      updatetool --version              (or)
      updatetool -v

Device Version:=
      updatetool --dev=<BIOS|CPU_CPLD|System_CPLD|ALL> --device_version (or)
      updatetool -D <devname | ALL> -V

Update the Device:=
      updatetool --dev=<BIOS|CPU_CPLD|System_CPLD> --update --file=<filename> (or)

```

```
updatetool -D <devname> -U -e filename
```

Usage:=

```
-h, --help           Show the help text
-v, --version        Display version
-D, --dev=           device name
-U, --update         update device
-V, --device_version show device version
-e, --file=         device file
-f, --config=       To specify the location of the config file e.g. /etc/dn/diag/
<file_name>
```

Device function list:

```
Device name: BIOS
  Support operation:
    update device
    show device version
```

```
Device name: CPU_CPLD
  Support operation:
    update device
    show device version
```

```
Device name: System_CPLD
  Support operation:
    update device
    show device version
```

Output

```
root@dellemc-diag-os:~# updatetool --device_version --dev=CPU_CPLD
```

```
CPU_CPLD version:
x.x
```

```
root@dellemc-diag-
```

BIOS update: N2200-ON or N3200-ON Series Only

```
root@dellemc-diag-os:~# ifconfig eth0 xxx.xxx.xx.xx
```

```
root@dellemc-diag-os:~# route add default gw xxx.xxx.xx.xxx
```

```
root@dellemc-diag-os:~# ping xx.xx.x.xx
```

```
PING xx.xx.x.xx (10.14.1.92) 56(84) bytes of data.
64 bytes from xx.xx.x.xx: icmp_seq=1 ttl=56 time=224 ms
64 bytes from xx.xx.x.xx: icmp_seq=2 ttl=56 time=224 ms
64 bytes from xx.xx.x.xx: icmp_seq=3 ttl=56 time=224 ms
64 bytes from xx.xx.x.xx: icmp_seq=4 ttl=56 time=224 ms
64 bytes from xx.xx.x.xx: icmp_seq=5 ttl=56 time=224 ms
^C
```

```
--- xx.xx.x.xx ping statistics ---
```

```
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
```

```
rtt min/avg/max/mdev = 224.076/224.196/224.369/0.530 ms
```

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

```
root@dellemc-diag-os:~# scp <name>@xx.xx.x.xx://work//archive1/sw/BIOS/<platform>_<platform>/<platform>-
BIOS-x.xx.x-x-x/<platform>-BIOS-x.xx.x-x-x.bin .
```

```
Password:
```

```
<platform>-BIOS-x.xx.x-x-
```

```
x.bin
```

```
100% 32MB 468.1KB/s 01:10
```

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

```
root@dellemc-diag-os:~# updatetool --dev=BIOS --update --file=<platform>-BIOS-x.xx.x-x-x.bin
```

```
Read device:BIOS backup file
```

```
Warning: BIOS region SMM protection is enabled!
```

```
Warning: Setting Bios Control at 0xdc from 0xaa to 0x89 failed.
```

```
New value is 0xaa.
```

```
Warning: SPI Configuration Lockdown activated.
```

```
FREG2: Warning: Management Engine region (0x00001000-0x01595fff) is locked.
```

```

Not all flash regions are freely accessible by flashrom. This is most likely
due to an active ME. Please see http://flashrom.org/ME for details.
PR0: Warning: 0x01860000-0x01e6efff is read-only.
PR1: Warning: 0x01e70000-0x01ffffff is read-only.
Enabling hardware sequencing because some important opcode is locked.
Reading flash... done.
Warning: BIOS region SMM protection is enabled!
Warning: Setting Bios Control at 0xdc from 0xaa to 0x89 failed.
New value is 0xaa.
Warning: SPI Configuration Lockdown activated.
FREG2: Warning: Management Engine region (0x00001000-0x01595fff) is locked.
Not all flash regions are freely accessible by flashrom. This is most likely
due to an active ME. Please see http://flashrom.org/ME for details.
PR0: Warning: 0x01860000-0x01e6efff is read-only.
PR1: Warning: 0x01e70000-0x01ffffff is read-only.
Warning: BIOS region SMM protection is enabled!
Warning: Setting Bios Control at 0xdc from 0xaa to 0x89 failed.
New value is 0xaa.
Warning: SPI Configuration Lockdown activated.
FREG2: Warning: Management Engine region (0x00001000-0x01595fff) is locked.
Not all flash regions are freely accessible by flashrom. This is most likely
due to an active ME. Please see http://flashrom.org/ME for details.
PR0: Warning: 0x01860000-0x01e6efff is read-only.
PR1: Warning: 0x01e70000-0x01ffffff is read-only.
Erase current image
Warning: BIOS region SMM protection is enabled!
Warning: Setting Bios Control at 0xdc from 0xaa to 0x89 failed.
New value is 0xaa.
Warning: SPI Configuration Lockdown activated.
FREG2: Warning: Management Engine region (0x00001000-0x01595fff) is locked.
Not all flash regions are freely accessible by flashrom. This is most likely
due to an active ME. Please see http://flashrom.org/ME for details.
PR0: Warning: 0x01860000-0x01e6efff is read-only.
PR1: Warning: 0x01e70000-0x01ffffff is read-only.
Enabling hardware sequencing because some important opcode is locked.
Erasing and writing flash chip... Erase/write done.
Write image to BIOS
Warning: BIOS region SMM protection is enabled!
Warning: Setting Bios Control at 0xdc from 0xaa to 0x89 failed.
New value is 0xaa.
Warning: SPI Configuration Lockdown activated.
FREG2: Warning: Management Engine region (0x00001000-0x01595fff) is locked.
Not all flash regions are freely accessible by flashrom. This is most likely
due to an active ME. Please see http://flashrom.org/ME for details.
PR0: Warning: 0x01860000-0x01e6efff is read-only.
PR1: Warning: 0x01e70000-0x01ffffff is read-only.
Enabling hardware sequencing because some important opcode is locked.
Erasing and writing flash chip... Erase/write done.
Verifying flash... VERIFIED.
write BIOS image success
Update BIOS image success
root@dellemc-diag-os:~#

```

CPLD update: N2200-ON or N3200-ON Series Only

```

root@dellemc-diag-os:~# ifconfig eth0 xxx.xxx.xx.xx
root@dellemc-diag-os:~# route add default gw xxx.xxx.xx.xxx
root@dellemc-diag-os:~#
root@dellemc-diag-os:~# ping xx.xx.xxx.xxx
PING 10.16.127.142 (10.16.127.142) 56(84) bytes of data.
64 bytes from 10.16.127.142: icmp_seq=1 ttl=62 time=0.116 ms
64 bytes from 10.16.127.142: icmp_seq=2 ttl=62 time=0.127 ms
64 bytes from 10.16.127.142: icmp_seq=3 ttl=62 time=0.099 ms
64 bytes from 10.16.127.142: icmp_seq=4 ttl=62 time=0.097 ms
64 bytes from 10.16.127.142: icmp_seq=5 ttl=62 time=0.099 ms
64 bytes from 10.16.127.142: icmp_seq=6 ttl=62 time=0.096 ms
^C
--- xx.xx.xxx.xxx ping statistics ---
24 packets transmitted, 24 received, 0% packet loss, time 23587ms
rtt min/avg/max/mdev = 0.093/0.106/0.140/0.016 ms

```

```

root@dellemc-diag-os:~#
root@dellemc-diag-os:~# scp <name>@xx.xx.xxx.xxx://sites/maa/work/tftpboot/users/<name>/
<platform>_cpu_rev_00_03.vme .
Password:
<platform>_cpu_rev_00_03.vme
100% 262KB 262.3KB/s 00:00
root@dellemc-diag-os:~#
root@dellemc-diag-os:~# updatetool --dev=CPU_CPLD --update --file=_cpu_rev_00_03.vme
Write image to CPU_CPLD
Lattice Semiconductor Corp.

ispVME(tm) V12.2 Copyright 1998-2011.

For daisy chain programming of all in-system programmable devices

TDI:41,TCK:43,TMS:42,WE:7,TRST:-1,TDO:44,SelPin:-1, Freq:1500
g_CoresiIspPins Init= 0 g_SussiIspPins Init= 0 g_WEAssertLevel= 0
TDI:41,TCK:43,TMS:42,WE:7,TRST:-1,TDO:44,SelPin:-1, Freq:1500
g_CoresiIspPins Init= 0 g_SussiIspPins Init= 0 g_WEAssertLevel= 0
TDI:41,TCK:43,TMS:42,WE:7,TRST:-1,TDO:44,SelPin:-1, Freq:1500
g_CoresiIspPins Init= 0 g_SussiIspPins Init= 0 g_WEAssertLevel= 0
Processing virtual machine file (n22xx_cpu_rev_00_03.vme).....

Diamond Deployment Tool 3.10
CREATION DATE: Tue Jan 08 13:11:21 2019

+====#+
| PASS! |
+====#+

write CPU_CPLD image success
Update CPU_CPLD image success
root@dellemc-diag-os:~#

```

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

Restore factory defaults

To restore your system factory defaults, reboot the system to ONIE: Uninstall OS mode.

If it is not possible to restore your factory defaults with the installed OS, reboot the system from the Grub menu and select `ONIE: Rescue`. ONIE Rescue bypasses the installed OS and boots the system into ONIE until you reboot the system. After ONIE Rescue completes, the system resets and boots to the ONIE console.

 **CAUTION: Restoring factory defaults erases any installed OS and requires a long time to erase storage.**

1. Restore the factory defaults on your system from the Grub menu using the `ONIE: Uninstall OS` command.

To select which entry is highlighted, use the up and down arrow keys.

```
GNU GRUB  version 2.02~beta2+e4a1fe391

+-----+
| ONIE: Install OS          |
| ONIE: Rescue              |
| *ONIE: Uninstall OS      |
| ONIE: Update ONIE       |
| ONIE: Embed ONIE        |
| EDA-DIAG                 |
|                           |
|                           |
|                           |
+-----+
```

2. Press **ENTER** to activate the console.
3. Return to the default ONIE settings using the `onie-uninstaller` command.

```
ONIE:/ # onie-uninstaller
Erasing internal mass storage device: /dev/sda4 (32MB)
  Percent complete: 100%
Erase complete.
Deleting partition 4 from /dev/sda
Erasing internal mass storage device: /dev/sda5 (300MB)
  Percent complete: 100%
Erase complete.
Deleting partition 5 from /dev/sda
Erasing internal mass storage device: /dev/sda6 (300MB)
  Percent complete: 100%
Erase complete.
Deleting partition 6 from /dev/sda
Erasing internal mass storage device: /dev/sda7 (12461MB)
  Percent complete: 100%
Erase complete.
Deleting partition 7 from /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
Uninstall complete. Rebooting...
ONIE:/ # discover: Rescue mode detected. No discover stopped.
Stopping: dropbear ssh daemon... done.
Stopping: telnetd... done.
Stopping: syslogd... done.
Info: Unmounting kernel filesystems
The system is going down NOW!
Sent SIGTERM to all processes
Sent SIGKILL tosd 4:0:0: [sda] Synchronizing SCSI cache
Restarting system.
machine restart
```

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, known as a luggage tag, or 11-digit express service code of your switch and click **Submit**.
- To view the chassis service tag or express service code, pull out the tag or enter the `show chassis` command from the CLI.
- To receive more technical support, click **Contact Us**. On the Contact Information web page, click **Technical Support**.

To access switch documentation, go to www.dell.com/support/.

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.