# Dell EMC Open Networking Troubleshooting Guide

March 2019

DELL EMC

## Notes, Cautions, and Warnings

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

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

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



2019 - March

Rev. A05

# Contents

# 1

# About this guide

This guide provides site preparation recommendations, step-by-step procedures for rack mounting and desk mounting, inserting optional modules, and connecting to a power source.

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

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

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


**Figure 1. Class 1 Laser Product ID Tag**

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

Topics:



## Information symbols

This book uses the following information symbols:

**NOTE: The Note icon signals important operational information.**

**CAUTION: The Caution icon signals information about situations that could result in equipment damage or loss of data.**

**WARNING: The Warning icon signals information about hardware handling that could result in injury.**

**WARNING: The ESD Warning icon requires that you take electrostatic precautions when handling the device.**

## Related documents

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

- *OS10 User Guide*
- *Dell EMC Command Line Reference Guide*
- *Dell EMC Configuration Guide*
- *Dell EMC Getting Started Guide*
- *Dell EMC Installation Guide*

- *Dell EMC Release Notes*
- *OS10 Release Notes*

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

# 2

# ONIE diagnostics

This chapter describes system diagnostics and troubleshooting. After running the diagnostic tools, 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: The troubleshooting package includes a README file that lists the tools version and the overall troubleshooting package version. For more information, see this README file.**

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

This system uses the following troubleshooting tools:

- **Power-On Self Test (POST) diagnostic** — Automatically runs during the system startup at the BIOS or U-boot 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.
- **Extended diagnostic application (EDA)** — Tests the hardware for system failures. This diagnostic tool is on-demand. EDA verifies platform-specific hardware. There are options to perform diagnostics from a Quick Test to a thorough Intrusive test. If a test fails, you can stop or continue boot-up. If you select the `halt-on failure` option, EDA testing does not continue. If you do not select the `halt-on failure` option, EDA testing continues. Test results are saved in a user-defined storage area. There is no physical alarm indication.

**NOTE: To test your hardware, Dell EMC strongly recommends using the EDA tool.**

**NOTE: EDA runs in the ONIE environment, not in the networking operating system. You must be at the ONIE prompt to run EDA.**

# ONIE expansion

To view all the ONIE commands available, 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
```

# Command-line interface options

Each diagnostic tool has the following options:

| Command | Description |
|---|---|
| **–h** | Help topics. Use help to find software-specific tools. |
| **test** | Tests against the preconfigured test file. |

Topics:

# System information

To view your S4810-ON or N1108EP-ON system information; for example, the model, part number, serial number, and service tag, follow these steps.

1 Reboot your system and enter U-Boot mode.

2 Enter the `sys_eeprom` command.

**Example of the `sys_eeprom` Command**

```
dell_<platform>_on-> sys_eeprom
TlvInfo Header:
   Id String:    TlvInfo
   Version:      1
   Total Length: 73
TLV Name             Code Len Value
-------------------- ---- --- -----
MAC Addresses        0x2A   2 65
Base MAC Address     0x24   6 00:E0:0C:02:01:FD
Vendor Name          0x2D   4 Dell
Product Name         0x21   8 <platform>
Part Number          0x22  10 7590009602
Serial Number        0x23  13 HADL127B20077
Manufacturer         0x2B   1 1
Service Tag          0x2F   2 123A1B2
Label Revision       0x27   3 A00
CRC-32               0xFE   4 0x4AF6A929
Checksum is valid.
```

# Boot processes

After the BIOS or U-Boot hardware verifications, POST tests run to verify the CPU and memory prior to booting the system software.

**NOTE: The N1108EP-ON switch is a U-Boot-based platform.**

After POST testing, there are three additional types of diagnostic tools you can use for testing your system.

- Manual diagnostic boot process — To run additional testing, manually download and run the EDA tool. The EDA tool reports and logs pass/fail results.
- ONIE with EDA — EDA is installed; you do not have to manually download the tool. Select the diagnostic option at boot-up. You can run this tool without a management interface.
- Autorun EDA — EDA is installed; you do not have to manually download the tool. Select the diagnostic option at boot-up. You can run this tool without a management interface. The system always launches EDA in Quick Test mode to verify the hardware components before loading the software. If there is a failure at boot-up, based on the EDA configuration, the software may or may not continue the boot process.

# POST

POST diagnostics verifies system memory before the software loads. Test configuration parameters are saved in CMOS for the next boot-up.

# EDA, Quick Test Mode

Quick Test mode runs basic device access tests for the system hardware to verify that the device is active and responding.

In Quick Test mode, the EDA tool quickly tests if the hardware components are accessible. It confirms that the components respond to read access and in some cases, simple write access. Tests are read-only and non-destructive (except the `memtool` command, which does allow read/write operations).

# Capturing Support Data from ONIE

To capture support data from ONIE, use the following commands.

1 Capture support data to the screen.
**ONIE:/ # dmesg**

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

# Changing the Default Grub Boot Entry

To view or set the default Grub boot entry, use the following command.
The `onie-boot-mode` command has two options —`l` (the default) and —`o`. The Grub boot default is to show 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
- `diag` — 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 is:

- Lists the current default entry. This is the default.

# Restoring to Factory Defaults

If you need to restore the S4810-ON or N1108EP-ON factory defaults, reboot the system to ONIE Rescue using the `run onie-rescue` or `run onie-diag` commands. If it is not possible to do this with the operating system you installed, reboot the system and hit any key to stop autoboot.

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

1 To restore the S4810–ON or N1108EP-ON factory defaults, run one of the following commands:

U-boot mode

`run onie_rescue` OR `run onie_diag`

ONIE Rescue bypasses the installed operating system and boots the system into ONIE until you reboot the system.

2 Press ENTER to activate the console.

**Example of the Optional `ONIE-uninstaller` Command**

After ONIE Rescue completes, the system resets and boots to the ONIE console.

**NOTE: Only use the optional `onie-uninstaller` command if you want to remove all the network operating software on your system except for ONIE. This command removes any installed network operating system.**

```
ONIE:/ # onie-uninstaller
    Erasing unused NOR flash region
    Erasing 128 Kibyte @ 20000 - 100% complete.
    Erasing internal mass storage device: /dev/mmcblk0 (7832MB)
    Percent complete: 100%
```

# Diagnostic package

To download the diagnostic package on an S4810–ON, S6000–ON, or N1108EP-ON platform, follow these steps:

**NOTE: Before you begin, go to http://www.dell.com/support and download the diagnostic package. You need your Dell EMC support access account to download the package.**

**NOTE: These steps only apply to the S4810–ON, S6000–ON, or N1108EP-ON platforms.**

1 Enter the `onie-discovery-stop` command to stop the ONIE discovery mode.

2 Assign an IP address to the management interface and verify the network connectivity.

```
ONIE:/ # ifconfig eth0 xx.xx.xx.xx/x
ONIE:/ #
ONIE:/ #
ONIE:/ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 90:B1:1C:F4:9C:76
          inet addr:xx.xx.xx.xx  Bcast:xx.xx.xx.xx  Mask:xx.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

ONIE:/ # ping xx.xx.xx.xx
PING xx.xx.xx.xx (xx.xx.xx.xx): 56 data bytes
64 bytes from xx.xx.xx.xx: seq=0 ttl=62 time=1.357 ms
64 bytes from xx.xx.xx.xx: seq=1 ttl=62 time=0.577 ms
^C
```

3 Upgrade the DIAG Installer. Again, boot to ONIE Rescue mode and install onie diag installer.

**NOTE: The command `onie-nos-install`, shown in bold, is not available in S4810-ON ONIE version 1.0.0.1; instead use the `install_url ONIE:/bin # install_url` command.**

```
ONIE:/ # onie-nos-install tftp://xx.xx.xx.xx/ON-DIAG/<platform>/JUL-08-2014/diag-installer-
powerpc-dell_<platform>_on_p2020-r0.bin
Stopping: discover... done.
Info: Fetching tftp://xx.xx.xx.xx/ON-DIAG/<platform>/JUL-08-2014/diag-installer-powerpc-
dell_<platform>_on_p2020-r0.bin ...
ON-DIAG/S4810/JUL-08 100% |*******************************| 1361k 0:00:00 ETA
ONIE: Executing installer: tftp://xx.xx.xx.xx/ON-DIAG/<platform>/JUL-08-2014/diag-installer-
powerpc-dell_<platform>_on_p2020-r0.bin
Verifying image checksum ... OK.
Preparing image archive ...
Preparing image archive ...sed -e '1,/^exit_marker$/d' /installer | tar xf - OK.
Diag Installer: platform: powerpc-dell_<platform>_on_p2020-r0
Erasing block: 128/128 (100%)
Writing kb: 16376/16384 (99%)
Verifying kb: 16376/16384 (99%)
ONIE:/ # umount: can't remount rootfs read-only
The system is going down NOW!
Sent SIGTERM to all processes
Sent SIGKILL toRestarting system.
Reset via the platform CPLD
```

4 Start ONIE diagnostics. To start the ONIE diagnostics for the S6000–ON, use the `onie-diag` option from the ONIE menu. To start the ONIE diagnostics for the S4810–ON or N1108EP-ON, use the following procedure:

a Set the u-boot environment *onie_boot reason* using `ONIE:/ # onie-set-env onie_boot_reason diag` from the ONIE # prompt or using `dell_<platform>_on > setenv onie_boot_reason diag` from the u-boot prompt.

b Reboot the system to launch and run the ONIE diagnostics.

For more information and output examples, see edatool.

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

# Troubleshooting issues

To help you solve an issue, use the following troubleshooting solutions.

**Table 1. Troubleshooting Issues and Solutions**

| Problem | Description and Solution |
|---|---|
| • A tool indicates a device as failing, but I do not believe this is correct.<br>• A tool indicates a device as passing, but I do not believe this is correct. | • If the configuration file for the particular tool was somehow corrupted, the results may be non-deterministic or inaccurate. Download the diagnostic package again; this ensures a clean copy of the configuration files. Then, re-run the tool. |
| I need to verify the version of a specific EDA tool. | • All versions of the tools loaded from the diagnostic package are listed in the `/diag/README` file. When you run `edatool`, the output from the README file is output to the console and the `logfile /mnt/diag.log` file.<br>• Any updates to tools are in the form of a new diagnostic package, which contains the corresponding README file. |
| The `storagetool` command runs and indicates a failure, but I do not believe this is correct. | • To run tests, the storage tool requires that you mount the storage device. To verify that there are mounted devices, use the `mount` command. If the device is mounted, the results indicate a problem with the physical device. |

| Problem | Description and Solution |
|---|---|
| The `pltool` command ran, reports a failure, and refers to a "mismatch". What does that mean? | • The `pltool` command compares the firmware versions of the device(s) to the expected latest revision. This message indicates that the firmware is not the most current. Contact your Dell EMC support representative. |
| The system is not allowing OS installation. | Run the following command:<br>`ONIE#onie-boot-mode -o rescue`<br>then follow the normal installation instructions. |
| **For the S6000–ON platform only.**<br>The `Fantool` reports a failure, but the fans seem to be working correctly. | • This is a bug in `fantool` in the current S6000–ON EDA. The `fantool` reads the fan information and compares it to values expected when the fan is initialized and is in the power-up state. |

# Troubleshooting tools

This section describes the diagnostic tools that provide debug and hardware tests.

To use the troubleshooting tools manually, you must be at the ONIE prompt and navigate to `/mnt/diag/`. If you are not already at the ONIE prompt, reboot your system to Rescue mode. Refer to your network operating software documentation for the procedure to reboot your system to Rescue mode.

**NOTE: EDA Quick Test mode only uses the tools in Access Only method for minimal system verification. EDA Extended mode uses the full system for debugging and verification.**

The EDA tool is a script-based execution of the other troubleshooting tools. The configuration file is a command-line execution for all tools to run in order.

**NOTE: After running the troubleshooting tools, the system reboots to the ONIE prompt. To return to your network operating software prompt, issue the `reboot` command.**

# edatool

The EDA tool (`edatool`) executes all of the other tools for testing and troubleshooting.

**NOTE: For troubleshooting your system, Dell EMC strongly recommends using the EDA tool and not individual tool commands.**

The `edatool` is script-based and is easily extended or narrowed to meet your requirements.

**Example of the `edatool` Output**

```
DIAG:/ # edatool
****************************
*  Diagnostics Application  *
****************************
DellEmc Diag edatool version x.x, package x.xx.x.x-x 2019/01/07
Writing data to block 127 at offset 0x1fc0000
DellEmc Diag cpldupgradetool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag cputool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag eepromtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag gpiotool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag i2ctool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag ledtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag memtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag nputool - version x.x sdk-x.x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag nvramtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag opticstool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag pltool - version x.x package x.xx.x.x-x 2019/01/07
```

```
DellEmc Diag poetool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag rtctool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag smbiostool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag storagetool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag temptool - version x.x package x.xx.x.x-x 2019/01/07
Testing I2C devices:

Checking I2C devices on bus 0:

+ Checking Thermal             0x4a  ..... Passed

+ Checking System CPLD         0x21  ..... Passed

+ Checking ID EEPROM Block 1   0x50  ..... Passed

+ Checking ID EEPROM Block 2   0x51  ..... Passed

+ Checking ID EEPROM Block 3   0x52  ..... Passed

+ Checking ID EEPROM Block 4   0x53  ..... Passed

+ Checking ID EEPROM Block 5   0x54  ..... Passed

+ Checking ID EEPROM Block 6   0x55  ..... Passed

+ Checking ID EEPROM Block 7   0x56  ..... Passed

+ Checking ID EEPROM Block 8   0x57  ..... Passed

I2C Devices: Overall test results --------------- >>> Passed
Testing Temp sensor  devices:
+ Checking [Thermal sensor]    = 43.0 C ............. Passed
Temp Sensors: Overall test results -------------- >>> Passed
PL Tool test:


System CPLD: LED Control Reg Reg Addr: 0x43 ...........Passed


Overall Test Results:  Passed

Show Optics in System
Port # Name         Status  Type Part Number  Rev Serial Number
------ ------------ ------ ------- ---------- --- ---------------
   1          SFP 1 PRESENT None 616740000    C   CN0C6Y7M65Q8WP3
   2          SFP 2 PRESENT None 616740000    C   CN0C6Y7M65Q8WP3
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 ......................................... Passed
Mounted Filesystem Devices:
DMA pool size: 8388608
AXI unit 0: Dev 0x8443, Rev 0x11, Chip BCM53443_B0, Driver BCM56160_A0
sysconf_probe successful
global_sal_config successful
***  1 BCM devices are detected
SOC unit 0 attached to PCI device BCM53443_B0
Current mode is now ESW
```

```
I2C: detected 0 devices
Diag NPU initialization over
         Test link_status_test for NPU 0 ................... Passed
       Test snake_traffic_test for NPU 0 ................... Passed
            Test prbs_mac_test for NPU 0 ................... SKIPPED <<<---
            Test prbs_ext_test for NPU 0 ................... SKIPPED <<<---

NPU tests .....................................  Passed

EDA: Overall test results ------------------------------ >>> Passed
```

The following shows the `edatool` extended test output.

**DIAG:/ # edatool --config=/etc/dn/diag/default_eda_extended_script.cfg**

```
*****************************
*  Diagnostics Application  *
*****************************
DellEmc Diag edatool version x.x, package x.xx.x.x-x 2019/01/07
Writing data to block 127 at offset 0x1fc0000
DellEmc Diag cpldupgradetool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag cputool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag eepromtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag gpiotool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag i2ctool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag ledtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag memtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag nputool - version x.x sdk-6.5.5 package x.xx.x.x-x 2019/01/07
DellEmc Diag nvramtool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag opticstool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag pltool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag poetool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag rtctool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag smbiostool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag storagetool - version x.x package x.xx.x.x-x 2019/01/07
DellEmc Diag temptool - version x.x package x.xx.x.x-x 2019/01/07
Testing I2C devices:

Checking I2C devices on bus 0:

+ Checking Thermal             0x4a  ..... Passed

+ Checking System CPLD         0x21  ..... Passed

+ Checking ID EEPROM Block 1   0x50  ..... Passed

+ Checking ID EEPROM Block 2   0x51  ..... Passed

+ Checking ID EEPROM Block 3   0x52  ..... Passed

+ Checking ID EEPROM Block 4   0x53  ..... Passed

+ Checking ID EEPROM Block 5   0x54  ..... Passed

+ Checking ID EEPROM Block 6   0x55  ..... Passed

+ Checking ID EEPROM Block 7   0x56  ..... Passed

+ Checking ID EEPROM Block 8   0x57  ..... Passed

I2C Devices: Overall test results ---------------------- >>> Passed
Testing Temp sensor  devices:
+ Checking [Thermal sensor]    = 44.0 C ....................... Passed
Temp Sensors: Overall test results -------------------- >>> Passed
PL Tool test:


System CPLD: LED Control Reg Reg Addr: 0x43 ..............Passed
```

```
Overall Test Results:  Passed

Show Optics in System
Port #  Name           Status  Type    Part Number      Rev  Serial Number
------  ------------   ------  ------- ---------------  ---  ---------------
   1          SFP 1   PRESENT None    616740000         C    CN0C6Y7M65Q8WP3
   2          SFP 2   PRESENT None    616740000         C    CN0C6Y7M65Q8WP3
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 ......................................... Passed
Mounted Filesystem Devices:
DMA pool size: 8388608
AXI unit 0: Dev 0x8443, Rev 0x11, Chip BCM53443_B0, Driver BCM56160_A0
sysconf_probe successful
global_sal_config successful
***  1 BCM devices are detected
SOC unit 0 attached to PCI device BCM53443_B0
Current mode is now ESW
I2C: detected 0 devices
Diag NPU initialization over
         Test link_status_test for NPU 0 ..................... Passed
       Test snake_traffic_test for NPU 0 ..................... Passed
            Test prbs_mac_test for NPU 0 ..................... SKIPPED <<<---
            Test prbs_ext_test for NPU 0 ..................... SKIPPED <<<---

NPU tests ....................................  Passed

Running Extended Tests:

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 12/12/2014, RTC time to 11:21:23.
Passed
Testing Storage Devices ..................... Passed
LED Test Started... Will take few mins to complete.


Overall LED test result ====>> Passed

EDA: Overall test results ---------------------------- >>> Passed
```

The EDA tool tests the platforms using the **default_eda_script.cfg configuration** file. This script dictates how EDA runs and which test runs first or multiple times. Each line of the script is the actual prompt command line including parameters.

# pcitool

The PCI tool (`pcitool`) allows testing of the PCI devices.

In EDA Quick Test mode, `pcitool` completes a simple check on the PCI bus. The tool scans the PCI bus for all drivers and functions and writes the configuration registers to the configuration file.

**Example of the `pcitool` Output**

```
Syntax: ./pcitool <option>
                                    -h := show this help
                                  scan := scan all PCI devices
                                   all := scan and show all config data
                                  test := test using the default PCI test config file
                 show <bus# dev# func#> := show config data for a specific bus:dev.func
   read <bus# dev# func# offset count> := read 8-bit config register for bus:dev.func
   write <bus# dev# func# offset data> := write 8-bit config register for bus:dev.func
```

**Example of the `pcitool` Configuation File Output**

```
# more default_pci_list.cfg
Bus:Dev.Fn=00:00.0 ID=0c738086 NOT LISTED
  0c738086 00000007 06000002 00000000 00000000 00000000 00000000 00000000
  00000000 00000000 00000000 00008086 00000000 00000000 00000000 00000000
  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  00000000 00000000 00000000 00000000 01080000 00000000 00020fb1 00000000
Bus:Dev.Fn=00:01.0 ID=0c468086 PCIe port
  0c468086 00100147 06040002 00010010 ff760004 00000000 00010100 000000f0
…
```

# i2ctool

The `i2ctool` allows testing of the devices on the i2c bus.

In EDA Quick Test mode, `i2ctool` scans busses and identifies all of the devices. If a device is behind a MUX, `i2ctool` looks for devices through the MUX and, if present, a second-level MUX as well.

**Example of the `i2ctool` Ouput**

```
Syntax: ./i2ctool <option>
     -h := show this help
   test := test <user_i2c_file.cfg> the pre-programmed configuration
           # ./i2ctool test userI2cFile.cfg
   read := read I2C device with <bus> <dev> <address> <bytecount>
           # ./i2ctool read /dev/i2c-0 0x50 0x00 10
  write := write I2C device with  <bus> <dev> <address> <data0> ... <dataN>
           # ./i2ctool write /dev/i2c-0 0x50 0x00 0x0a 0x0b 0x0c
   scan := scan <bus_prefix> the I2C devices on the specified bus prefix
           # ./i2ctool scan /dev/i2c-
```

**Example of the `i2ctool` Configuration File Output**

```
# more gpio_00_i2c_devices.cfg
I2C devices found on bus #0: 10
Dev found @ 0x18,/dev/i2c-0,-,-,0x18,0x00,1
Dev found @ 0x30,/dev/i2c-0,-,-,0x30,0x00,1
Dev found @ 0x31,/dev/i2c-0,-,-,0x31,0x00,1
Dev found @ 0x32,/dev/i2c-0,-,-,0x32,0x00,1
Dev found @ 0x33,/dev/i2c-0,-,-,0x33,0x00,1
Dev found @ 0x3e,/dev/i2c-0,-,-,0x3e,0x00,1
```

```
Dev found @ 0x4d,/dev/i2c-0,-,-,0x4d,0x00,1
Dev found @ 0x50,/dev/i2c-0,-,-,0x50,0x00,1
Dev found @ 0x53,/dev/i2c-0,-,-,0x53,0x00,1
Dev found @ 0x69,/dev/i2c-0,-,-,0x69,0x00,1
I2C devices found on bus #1: 0
I2C devices found on bus #2: 2
Dev found @ 0x51,/dev/i2c-2,-,-,0x51,0x00,1
Dev found @ 0x59,/dev/i2c-2,-,-,0x59,0x00,1
```

# memtool

The memory tool (`memtool`) tests system memory.

EDA Quick Test mode only completes simple access (read) tests. You can use `memtool` to test static memory areas, such as L2Cache mapped as SRAM or DRAM on a memory mapped device.

The memory configuration file consists of lines that describe a region of memory and the tests performed on that memory region. Therefore, you can have multiple entries for a region of memory. All parameters are separated by a : character. The following describes the configuration file parameters.

| Parameter | Description |
|---|---|
| **Region Name:** | The region name referred to in all output. |
| **Start Address:** | The starting address for the region of memory in hexadecimal format (without the preceding 0x). If the operating software is defining how to manage memory, this is dynamic and you can use "-" for the start address and the memory uses malloc'd from the available system memory. |
| **Size:** | The size of the contiguous memory area (in bytes) in hexadecimal (without the preceding 0x). If memory is dynamic, use "-" for size and the tests use the maximum system memory. |
| **Access:** | The letter that describes how this memory is accessed:<br>b — byte (8 bits)<br>h — halfword (16 bits)<br>w — word (32 bits)<br>d — double-word (64 bits) |
| **Increment:** | How many bytes to increment to the next cell. |
| **ECC:** | Describes if ECC is supported — 0 or 1. |
| **Chunk:** | Describes how many kilo bytes (1024 bytes) are tested in one chunk. Tests multiple chunks across the memory region. |
| **Max Cache:** | The maximum cache size for this memory. |
| **Cacheline:** | The size of a cacheline. |
| **Iterations:** | The number of times to perform the tests. |
| **Test:** | The collection of bits that tell which test to perform on this region. The tests are performed in bit order. Some tests may not be performed due to time limitations and the purpose of the test (for example, *dim cache memory test* which is time consuming and destructive to data). To run an excluded test, you must specifically request the test. For example, to run all tests including the *dim cache memory* test, which is a 0x800, set the tests to `fff`.<br>A-1 — run all available tests. |

| Parameter | Description |
|---|---|
| Descriptive Device: | The descriptive device (for example, SPD in the case of a dimm), is described in one comma-separated field of four parameters: address, type, start, and bytes. |
| Device Path: | The path to the device driver (for example, `/dev/i2c-0` for a PD on the i2c bus 0). |
| Address of Memory Description device (SPD): | Describes the memory organization (for example, dimm memory which has an SPD device) and the address of the device. For i2c devices, the address is presented in 7–bit hexadecimal format. |
| Type: | SPD |
| Range of bytes in the Description Device: | These two fields list the start and end registers to read descriptive entries for the device. |

**Example of the `memtool` Output**

```
Syntax: ./memtool <option>
                                  -h := show this help
           test [[all]|list|region#] := test using the MEM test config file
                               info  := display configuration info of device
                read [b|h|w] address := read the specified physical address
 write [b|h|w] address data [length] := write at the specified physical address
```

**Example of the `memtool` Configuration File Output**

```
//    Memory Configuration File
//
//    Example:
//    SystemRam:-:-:w:4:1:2800:0:0:1:-1:SPD:/dev/i2c-0:50:0,ff:
//    This describes the SystemRam which is dynamic in location and size.  It is accessed by
words //    and incremented addresses of 4 bytes. It is ECC covered, and has a max chunk of
10KB max
//    cache and cacheline size (unused at this time) are 0.  The tests will be performed once
on this
//    region, and the -1 denotes to run all tests, excluding dim cache memory test. The
Descriptive
//   device is a SPD on /dev/i2c-0 at address 0x50, and we read registers 0-255.
//
//    Note: a '-' address and size denotes a dynamic ram allocation
//    ======================Tests================================
//    -1  : all Tests Run
//     0h : No Address Test
//     1h : Address Read Test (Access)
//     2h : Address Read|Modify|Write|Verify
//     4h : Address walking 1's
//     8h : Address walking 0's
//     00h : No Data Test
//     10h : Data Read Test (Access)
//     20h : Data Read|Modify|Write|Verify
//     40h : Data walking 1's
//     80h : Data walking 0's
//    100h : Data walking 1's
//    200h : Data walking 0's
//    400h : patterns (00ff, ff00, 55aa, aa55)
//    800h : Cache (cacheKiller - Not Part of ALL Tests)

SystemRam:-:-:d:8:1:2800:-1:-1:1:-1:i2c:0x52,SPD,0,255
```

# pltool

The programmable logic tool (`pltool`) verifies access to the complex programmable logic devices (CPLD) and field programmable gate array (FPGA) and verifies versions.

The `pltool` generates its configuration file based on the platform database. The configuration file is generated with a specific version of devices in order to detect manufacturing misleads. The database holds all the versions and is updated when new versions are released.

The configuration file displays in tree format. The base is the chip that can have multiple registers and may or may not have bit descriptions and bit collection information. Each parameter in the tree is on an individual line separated by the "|" character.

**Example of the `pltool` Output**

```
Syntax: ./pltool <option>
                                        -h := show this help
                                      test := test using the test config file
                                      list := list devices and registers
      read [b|h|w] device offset [length]  := read the specified register
 write [b|h|w] device offset data [length] := write at the specified register
```

The configuration file displays in tree format. The base is the chip that can have multiple registers and may or may not have bit descriptions and bit collection information. Each parameter in the tree is on an individual line separated by the "|" character.

The following describes the C-row configuration file tree output.

| C-Row Parameter | Description |
|---|---|
| C | The row identifier. |
| Type | CPLD, FPGA, or TPM. |
| Address | The address of the device. For CPLD on an I2C bus, this is an 8–bit address. For PCI, this is the bus:dev:function in a packed 32–bit word, 8–bits each. For memory, this is the address. LPC is unknown. |
| Name | The text name for the device. |
| Interface | i2c, pci, mem, io, or lpc. |
| Bus | For devices on multiple busses, this indicates the bus number (for example, 0 for i2c indicates /dev/i2c-0. For PCI, this holds the bus-dev-func in a 32–bit value with each byte representing bus, device and functions in that order (for example, 0x020304 represents bus 02, device 03, and function 04). |
| Version Reg | The register that contains the version for the device. |
| Version Mask | The bits to use to check the version. |

The following describes the R-row configuration file tree output.

| R-Row Parameter | Description |
|---|---|
| R | The register identifier. |
| Address | The register addresses (offset) in the device. |
| Register Size | Describes how many bits the register contains (for example, 8, 16, or 32). |
| Register Mask | Lists the valid bits in this register. |
| Name | The text name of this register. |
| Access Perm | Access permissions: |

| R-Row Parameter | Description |
|---|---|
| | RO — Read only<br>RW — Read/Write<br>RC — Clear on Read<br>WO — Write only |
| Default Value | The default value of this register. |
| Testable | 1 — The register can be tested against the default value.<br>0 — The register is not testable. |
| Version | The version of the register. There can be multiple definitions of a register based on the version. When the test creates a configuration file from a device list with several versions, a specific version is requested and if the version requested is the last version prior to or equal to the requested version, it is put into the configuration file. |

The following describes the B-row configuration file tree output.

| B-Row Parameter | Description |
|---|---|
| B | The bit row identifier. |
| Bit Number(s) | This can be either a single bit number or a range starting with the highest bit number (for example, 7 or 7:3). |
| Name | The name of the bit. |
| Access | The access type of the bit; the same as the register definition. |
| Default Value | The default value of the bits. |

The following describes the I-row configuration file tree output.

| I-Row Parameter | Description |
|---|---|
| I | The identifier row descriptor. |
| Value | The value of the collection of bits. |
| Meaning | The meaning of the collection of bits. |

**Example of the `pltool` Configuration File Output**

```
# ./pltool
Programable Logic Tool
Syntax: ./pltool <option>
                                        -h := show this help
                                     test := test using the test config file
                                     list := list devices and registers
      read [b|h|w] device offset [length]  := read the specified register
 write [b|h|w] device offset data [length] := write at the specified register
```

**Example of the `pltool` Configuration File Tree Output**

```
# C - CHIP (Master | Slave -  Cpld or FPGA), Address, Name, Access
# R - Register, Offset, Mask, Name, RW , Default Val
# B - Bit(s), bitnum(s), Name, RW, Default Val
# I - Information on the bits
=====
C | CPLD | 0x31 | System CPLD | i2c | 0 | 0x00 | 0xf
R | 0x00 | 8 | 0xFF | Board Revision Reg | RO | 0x4A | 1 | 0x0
```

```
B |  7:4 | Board Stage | RO | 0x0
B |  3:0 | CPLD Revision | RO | 0x0
R | 0x01 | 8 | 0xFF | Software Reset Reg | RW | 0xFF | 0 | 0x0
B |  7 | Reserved | RW | 0x1
B |  6 | CFast Card Pres | RO | 0x1
B |  5 | CPU HRSTn | RW | 0x1
B |  4 | Super IO RST | RW | 0x1
B |  3 | PE_SATA_RST | RW | 0x1
B |  2 | PE_USB_RST | RW | 0x1
B |  1 | FORCE_RST | RW | 0x1
B |  0 | CPU_RST | RW | 0x1
=====
C | CPLD | 0x32 | Master CPLD | i2c | 0 | 0x01 | 0xf
R | 0x01 | 8 | 0xFF | Board Revision Reg | RW | 0x4C | 1 | 0x0
B |  7:4 | Board Stage | RO | 0x0
I |  4 | P2B-P2C Stage
I |  3 | P2A Stage
I |  2 | P1 Stage
I |  1 | P0 Stage
I |  0 | Testing Code
B |  3:0 | CPLD Revision | RW | 0x0
R | 0x02 | 8 | 0xFF | Power Enable Reg 1/2 | RO | 0x0 | 0 | 0x0
```

# gpiotool

The `gpiotool` verifies the state of the GPIO signals.

This test is more commonly used in ODA Offline and Online modes. `gpiotool` can also change the GPIO values for I/O devices and internal CPU GPIOs.

**Example of the `gpiotool` Output**

```
Syntax: ./gpiotool <option>
    -h := show this help
   set := set GPIO pin
          # set pin# value
   get := get GPIO pin value
          # get pin#
```

# storagetool

The storage tool (`storagetool`) tests the physical media and SMART status media, if supported.

`storagetool` is commonly used with ODA Offine testing.

**Example of the `storagetool` Output**

```
Syntax: ./storagetool <option>
                        -h := show this help
                      list := list devices
            test <device> := test devices (empty for all)
           smart <device> := get the smart status for a device (empty for all)
```

# psutool

The power supply tool (`psutool`) reports information about the power supplies. It can read and write the power supply parameters.

**NOTE: If you are missing a power supply unit, the `psutool` test reports a failure.**

**Example of the `psutool` Output (Preliminary Syntax)**

```
Syntax: ./psutool <option>
                              -h := show this help
```

```
    test [[all] | power supply] := test using the default config file
          read <psu> <register> := read the register on the Power Supply
  write <psu> <register> <value> := write the value into the Power Supply Register
```

# fantool

The fan tool (`fantool`) reports information about fan availability and fan speed. `fantool` also allows control of the fan speeds.

**NOTE: If you are missing a fan, the `fantool` test reports a failure.**

**Example of the `fantool` Output**

```
Syntax: ./fantool <option>
                                                  -h := show this help
                     test [[all] | fan controller] := test using the Fan Controller config file
                                                init := Initialize the fans to the default state
set <fan | all> [speed in RPM | low | mid | high] := sets the fan(s) to the speed
                                   get <fan | all> := gets the speed of the specified fan or all
fans in RPM
```

In the `fantool` configuration output, the first six rows describe the six fans in your system and the devices that control the fans. The second portion lists the speeds for the fans: low, medium, high, and default.

The first portion of the output lists the fan IDs, name, the device that controls the fan, the fan address on the I2C bus, and the instance of the fan control within the chip. Also included is a description of which fan tray the fan is located in. The last two values in the configuration file output are the speed resolution and the number of pulses per second.

**Example of the `fantool` Configuration File Output**

```
0 | Fan 1 | MAX6620 | 0x2a | 0 | Fan Tray 0 | 4 | 2
1 | Fan 2 | MAX6620 | 0x2a | 1 | Fan Tray 0 | 4 | 2
2 | Fan 3 | MAX6620 | 0x29 | 0 | Fan Tray 1 | 4 | 2
3 | Fan 4 | MAX6620 | 0x29 | 1 | Fan Tray 1 | 4 | 2
4 | Fan 5 | MAX6620 | 0x29 | 2 | Fan Tray 2 | 4 | 2
5 | Fan 6 | MAX6620 | 0x29 | 3 | Fan Tray 2 | 4 | 2
=====
0 | 5 | 2000 | 9000 | 19000 | 18000
```

# temptool

The temperature tool (`temptool`) allows access to the thermal sensors on the boards and devices.

Temperatures are reported in degrees C. `temptool` also allows you to set the temperature thresholds for error and monitoring.

**Example of the `temptool` Output**

```
Syntax: ./temptool <option>
    -h := show this help
  test := test the pre-programmed configuration
          # test <temperature_cfg_filename>
  show := show current temp device values for
          # show <temperature_cfg_filename>
```

# nvramtool

The NVRAM tool (`nvramtool`) allows the setting of the control bits for POST testing.

**Example of the `nvramtool` Output**

```
Syntax: ./nvramtool <option>
                                        -h := show this help
```

```
                                read := read nvram values at reg index
                      write <reg> <value> := write nvram value
```

# 3

# Dell EMC diagnostics

The following describes the Dell EMC diagnostics. These instructions apply to systems for which the ONIE diagnostics are not available.

# ONIE expansion

To view all the ONIE commands available, 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:

- S6000–ON system information
- S6000–ON factory defaults restore

# S6000–ON system information

To view your S6000–ON system information; for example, the model, part number, serial number, and service tag, follow these steps.

1 Boot into ONIE.

2 Enter the `onie-syseeprom` command.

**ONIE:/ # onie-syseeprom**

**Example of the `onie-syseeprom` Command**

```
TlvInfo Header:
   Id String:    TlvInfo
   Version:      1
   Total Length: 78
TLV Name             Code Len Value
-------------------- ---- --- -----
MAC Addresses        0x2A   2 129
Base MAC Address     0x24   6 00:05:33:6A:BF:4D
Vendor Name          0x2D   4 Dell
Product Name         0x21   8 S6000-ON
Part Number          0x22   6 08YWFG
Serial Number        0x23  12 DLCN13980015
Label Revision       0x27   3 A00
Manufacturer         0x2B   1 1
Service Tag          0x2F   2 ABC1AB2
Loader Version       0x29   8 x.xx.x.x
CRC-32               0xFE   4 0xC1EB87D1
Checksum is valid.
```

# S6000–ON factory defaults restore

If you need to restore the S6000–ON factory defaults, reboot the system to ONIE Rescue mode.

If it is not possible to do this with the operating system you installed, reboot the system and from Grub and select `ONIE: Rescue`.

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

ONIE Rescue bypasses the installed operating system and boots the system into ONIE until you reboot the system. After ONIE Rescue completes, the system resets and boots to the ONIE console.

1 Restore the S6000–ON factory defaults from Grub using the `ONIE: Rescue` command.

Use the up and down arrow keys to select which entry is highlighted. 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 (*) 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                 |
 | DELL EMC DIAG                    |
 |                                  |
 |                                  |
 |                                  |
 |                                  |
 |                                  |
 +----------------------------------+
```

2 Press ENTER to activate the console.

3 You can also use the `onie-uninstaller` command to return to the default ONIE settings.

```
ONIE:/ # onie-uninstaller
Erasing unused NOR flash region Erasing 128 Kibyte @ 20000 - 100% complete. Erasing internal
mass storage device: /dev/mmcblk0 (7832MB) Percent complete: 100%
```

# 4

# 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/manuals/.

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

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