



nvidia-healthmon User Guide

Version 2.2

NVIDIA Corporation

January 7, 2013



Contents

1	Overview	1
1.1	nvidia-healthmon goals	1
1.1.1	Beyond the scope of nvidia-healthmon	1
1.2	Dependencies	1
1.3	Supported products	2
2	Running nvidia-healthmon	3
2.1	Listing GPUs	3
2.2	Targeting a specific GPU	3
2.3	Modes	4
2.4	Verbose output	4
2.5	Other flags	4
3	Interpreting the result	5
3.1	nvidia-healthmon exit code	5
4	Configuring nvidia-healthmon	8
4.1	Configuration file contents	8

Chapter 1

Overview

nvidia-healthmon is the system administrator's and cluster manager's tool for detecting and troubleshooting common problems affecting NVIDIA®Tesla™GPUs in a high performance computing environment. nvidia-healthmon contains limited hardware diagnostic capabilities, and focuses on software and system configuration issues.

1.1 nvidia-healthmon goals

nvidia-healthmon is designed to:

1. Discover common problems that affect a GPU's ability to run a compute job including
 - (a) Software configuration issues
 - (b) System configuration issues
 - (c) System assembly issues, like loose cables
 - (d) A limited number of hardware issues
2. Provide troubleshooting help
3. Easily integrate into Cluster Scheduler and Cluster Management applications
4. Reduce downtime and failed GPU jobs

1.1.1 Beyond the scope of nvidia-healthmon

nvidia-healthmon is not designed to:

1. Provide comprehensive hardware diagnostics
2. Actively fix problems

1.2 Dependencies

This version of nvidia-healthmon depends on the NVIDIA Developer Display Driver version 304, found on the CUDA download page. <http://developer.nvidia.com/cuda-downloads> Additionally, NVIDIA Display Drivers versions 310, 313, and 316 are supported. These can be found on the NVIDIA driver download page. <http://www.nvidia.com/Drivers>

1.3 Supported products

nvidia-healthmon supports Tesla GPUs running on Linux operating systems.

NVIDIA®Tesla™Line:

- S1070
- S2050
- C1060
- C2050/70/75
- M2050/70/75/90
- X2070/90
- K10/K20

Chapter 2

Running nvidia-healthmon

Once unpackaged, nvidia-healthmon can be run from the command line.

Note that nvidia-healthmon will create a CUDA context on the GPU it is testing. In compute exclusive mode, this means that no other CUDA process will be able to create a CUDA context on the GPU. The results of nvidia-healthmon may also be affected by other CUDA processes. For example nvidia-healthmon may detect low bandwidth if another process is sending data over PCIe to any GPU on the system.

```
user@hostname
$ nvidia-healthmon
```

When no arguments are supplied, nvidia-healthmon will run with the default behavior on all supported GPUs.

2.1 Listing GPUs

nvidia-healthmon is able to list the GPUs installed on the system. This is useful to determine the PCI bus ID or device index needed in the next section. Note that the device index may change if the NVIDIA display driver is reloaded, or the system is rebooted.

```
user@hostname
$ nvidia-healthmon -L
```

For extended GPU information see the nvidia-smi tool:

```
user@hostname
$ man nvidia-smi
```

```
user@hostname
$ nvidia-smi -q
```

2.2 Targeting a specific GPU

nvidia-healthmon can target a single GPU or a set of GPU's. To target a specific GPU, run nvidia-healthmon using the '-i' or '-id' flag with the identifier of the GPU to be targeted. Identifiers are either:

1. A device index
2. A PCI bus ID
3. A GPU chip UUID
4. A GPU board serial number

NOTE: A board serial number will target all GPUs on the board.

```
user@hostname
$ nvidia-healthmon -i 0
```

```
user@hostname
$ nvidia-healthmon -i 0000:02:00.0
```

2.3 Modes

The default mode is quick mode. In quick mode a subset of tests are run, to quickly detect common problems. The '-q' or '--quick' flags allow quick mode to be explicitly requested.

The other mode is extended mode. In extended mode all available tests will be run. To run extended diagnostics run nvidia-healthmon using the '-e' or '--extended' flags.

```
user@hostname
$ nvidia-healthmon --extended
```

For more information about these modes see the nvidia-healthmon Best Practices Guide.

2.4 Verbose output

The default output of nvidia-healthmon will not report values for various metrics it has collected. The verbose flags, '-v' or '--verbose', can be used to print values like the pinned memory bandwidth and CUDA device query information. Additionally, verbose mode will provide information about why some tests were skipped.

Note that the output format of nvidia-healthmon may change in a later release.

2.5 Other flags

For a more complete list of the flags available, run nvidia-healthmon with the '-h'/'--help' or '-H'/'--verbose-help' flags.

```
user@hostname
$ nvidia-healthmon -h
```

Chapter 3

Interpreting the result

3.1 nvidia-healthmon exit code

nvidia-healthmon will terminate once it completes the execution diagnostics on all specified devices. An exit code of **zero** will be used when nvidia-healthmon runs successfully. A **non-zero** exit code indicates that there was a problem with the nvidia-healthmon run. This may be due to problems running the diagnostic: such as a missing configuration file, or invalid command line arguments, or problems with the local machine that nvidia-healthmon has detected. The output of the application must be read to determine what the exact problem was.

```
user@hostname
$ nvidia-healthmon -q

Loading Config: SUCCESS
Global Tests
  NVML Sanity: SUCCESS
  Tesla Devices Count: SUCCESS
  Global Test Results: 2 success, 0 errors, 0 warnings, 0 did not run

-----

GPU 0000:04:00.0 #1 : Tesla C2075 (Serial: 0425912072221)
  NVML Sanity: SUCCESS
  InfoROM: SKIPPED
  GEMINI InfoROM: SKIPPED
  ECC: SUCCESS
  CUDA Sanity
    Result: SUCCESS
  PCIe Maximum Link Generation: SUCCESS
  PCIe Maximum Link Width: SUCCESS
  PCI Seating: SUCCESS
  PCI Bandwidth: SKIPPED
  Device Results: 6 success, 0 errors, 0 warnings, 3 did not run

System Results: 8 success, 0 errors, 0 warnings, 3 did not run
```

An example of a successful run of nvidia-healthmon .

In the above example three of the tests did not run.

- The InfoROM test is an extended mode only test so it is skipped. To run the InfoROM test, extended mode must be used.
- The Gemini infoROM test was skipped because the Tesla C2075 only has a single GPU on the board.

- The PCIe bandwidth test was not run because the configuration file does not specify the bandwidth supported by the system. To run the bandwidth test, the configuration file must be edited.

Note that skipped tests will not affect the nvidia-healthmon exit code.

```
user@hostname
$ nvidia-healthmon -q
```

Loading Config: SUCCESS

Global Tests

NVML Sanity: SUCCESS

Tesla Devices Count: SUCCESS

Global Test Results: 2 success, 0 errors, 0 warnings, 0 did not run

GPU 0000:04:00.0 #1 : Tesla C2075 (Serial: 0425912072221)

NVML Sanity: SUCCESS

InfoROM: SKIPPED

GEMINI InfoROM: SKIPPED

ECC: SUCCESS

CUDA Sanity

Result: SUCCESS

PCIe Maximum Link Generation: SUCCESS

PCIe Maximum Link Width: SUCCESS

PCI Seating

ERROR: After enabling maximum performance mode, the current PCIe Link Width (8) does not match the expected maximum PCIe Link Width (16). This can indicate that this GPU is improperly seated.

An issue was detected with this GPU. This issue is usually caused by poor connection between the GPU and the system.

* Run 'nvidia-smi -q'. In some cases this will report a poorly connected power cable.

* Power down your system.

* Check that all power connectors are firmly attached (some GPUs require two power cables attached).

* Check that the power cable is not damaged. Symptoms of damaged power cables include exposed wiring and kinks (sharply creased region).

* Check that the power cable is attached to a working power supply.

* Rerun these diagnostics, using the same GPU, on system that is known to be working. A variety of system issues can cause diagnostic failure.

* Restart your system and install the latest NVIDIA display driver.

* Contact your OEM provider, to run further system diagnostics.

* Run 'nvidia-bug-report.sh' as the root user.

* Run 'nvidia-healthmon -v -e -l nvidia-healthmon-report.dump > nvidia-healthmon-report.log'

* Provide the files nvidia-bug-report.log.gz, nvidia-healthmon-report.dump, and nvidia-healthmon-report.log to your OEM to assist your OEM in resolving this issue.

Result: ERROR

PCI Bandwidth: SKIPPED

Device Results: 5 success, 1 errors, 0 warnings, 3 did not run

System Results: 7 success, 1 errors, 0 warnings, 3 did not run

WARNING: One or more tests didn't run. Read the output for details.

ERROR: One or more tests failed. Read the output for details.

An example of a failing run of nvidia-healthmon .

In the above example, nvidia-healthmon detected a problem with how the GPU was inserted into the system. nvidia-healthmon exited with a **non-zero** exit code. Additionally, the output provides a user readable description of what went wrong and a list of the steps the customer can take to solve the problem.

Chapter 4

Configuring nvidia-healthmon

While nvidia-healthmon will work out of the box without additional configuration, it is possible to configure the behavior and enable optional features. nvidia-healthmon is packaged with a sample configuration file, config.ini. This configuration file can be used to enable optional tests. By default, a test that is missing configuration information will be skipped.

The configuration file used can be specified on the command line. By default nvidia-healthmon will look in the current working directory for a file named config.ini.

```
user@hostname
$ nvidia-healthmon -c /path/to/your/config.ini
```

Note that a default config.ini file is provided with nvidia-healthmon . All of the available properties are listed in this file, with their descriptions.

4.1 Configuration file contents

This file is formatted in the standard ini file format.

There are two types of sections in the file. The first is the 'global' section, the second is a 'GPU name' section.

The global section contains expected system configuration information. For example, the 'devices.tesla.count' describes the number of GPUs on the system.

Any section name other than 'global' is a GPU name section. The GPU name section contains information about all GPUs with a given name. For example, the section [Tesla M2090] contains the configuration for all Tesla M2090 GPUs.

The two types of sections support different key value pairs. For instance, the key 'devices.tesla.count' is only allowed to be grouped underneath the 'global' category. The config.ini file nvidia-healthmon is packaged with contains the valid key value pairs for each section along with a brief description.

Listing 4.1 shows a sample configuration file for a system that contains 4 devices that are either Tesla™C2075's, or Tesla™C2070's.

Listing 4.1: A sample config.ini

```
[global]
devices.tesla.count = 4

[Tesla C2075]
bandwidth.warn = 1500
bandwidth.min = 100
pci.gen = 2
pci.width = 16
temperature.warn = 95
```

```
[Tesla C2070]
bandwidth.warn = 1500
bandwidth.min = 100
pci.gen = 2
pci.width = 16
temperature.warn = 90
```

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, GeForce, Tesla, and Quadro are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2007-2012 NVIDIA Corporation. All rights reserved.