

NAME

nvidia-smi – NVIDIA System Management Interface program

SYNOPSIS

nvidia-smi [OPTION1 [ARG1]] [OPTION2 [ARG2]] ...

DESCRIPTION

NVSMI provides monitoring information for each of NVIDIA's Tesla devices and each of its high-end Fermi-based and Kepler-based Quadro devices. It provides very limited information for other types of NVIDIA devices. See NVML documentation at <http://developer.nvidia.com/nvidia-management-library-nvml> for what features are supported on a particular device. The data is presented in either plain text or XML format, via stdout or a file. NVSMI also provides several management operations for changing device state.

Note that the functionality of NVSMI is exposed through the NVML C-based library. See the NVIDIA developer website for more information about NVML. Python and Perl wrappers to NVML are also available. The output of NVSMI is not guaranteed to be backwards compatible; NVML and the bindings are backwards compatible.

<http://developer.nvidia.com/nvidia-management-library-nvml/>

<http://pypi.python.org/pypi/nvidia-ml-py/>

<http://search.cpan.org/search?query=nvidia%3A%3Aml>

OPTIONS**GENERAL OPTIONS**

-h, --help

Print usage information and exit.

SUMMARY OPTIONS

-L, --list-gpus

List each of the NVIDIA GPUs in the system, along with their serial numbers or UUIDs. Tesla and Quadro GPUs from the Fermi and Kepler family report serial numbers, which match the ids physically printed on each board. GT200 Tesla products only support UUIDs, which are also unique but do not correspond to any identifier on the board. All other products report N/A.

QUERY OPTIONS

-q, --query

Display GPU or Unit info. Displayed info includes all data listed in the (*GPU ATTRIBUTES*) or (*UNIT ATTRIBUTES*) sections of this document. Some devices and/or environments don't support all possible information. Any unsupported data is indicated by a "N/A" in the output. By default information for all available GPUs or Units is displayed. Use the **-i** option to restrict the output to a single GPU or Unit.

[plus optional]

-u, --unit

Display Unit data instead of GPU data. Unit data is only available for NVIDIA S-class Tesla enclosures.

-i, --id=ID

Display data for a single specified GPU or Unit. The specified id may be the GPU/Unit's 0-based index in the natural enumeration returned by the driver, the GPU's board serial number, the GPU's UUID, or the

GPU's PCI bus ID (as domain:bus:device.function in hex). It is recommended that users desiring consistency use either UUID or PCI bus ID, since device enumeration ordering is not guaranteed to be consistent between reboots and board serial number might be shared between multiple GPUs on the same board.

-f FILE, --filename=FILE

Redirect query output to the specified file in place of the default stdout. The specified file will be overwritten.

-x, --xml-format

Produce XML output in place of the default human-readable format. Both GPU and Unit query outputs conform to corresponding DTDs. These are available via the **--dtd** flag.

--dtd

Use with **-x**. Embed the DTD in the XML output.

-d, --display

Display only selected information: MEMORY, UTILIZATION, ECC, TEMPERATURE, POWER, CLOCK, COMPUTE, PIDS, PERFORMANCE. Flags can be combined with comma e.g. "MEMORY,ECC". Doesn't work with **-u/--unit** or **-x/--xml-format** flags.

-l SEC, --loop=SEC

Continuously report query data at the specified interval, rather than the default of just once. The application will sleep in-between queries. Note that on Linux ECC error or XID error events will print out during the sleep period if the **-x** flag was not specified. Pressing Ctrl+C at any time will abort the loop, which will otherwise run indefinitely. If no argument is specified for the **-l** form a default interval of 5 seconds is used.

DEVICE MODIFICATION OPTIONS

[any one of]

-pm, --persistence-mode=MODE

Set the persistence mode for the target GPUs. See the (*GPU ATTRIBUTES*) section for a description of persistence mode. Requires root. Will impact all GPUs unless a single GPU is specified using the **-i** argument. The effect of this operation is immediate. However, it does not persist across reboots. After each reboot persistence mode will default to "Disabled". Available on Linux only.

-e, --ecc-config=CONFIG

Set the ECC mode for the target GPUs. See the (*GPU ATTRIBUTES*) section for a description of ECC mode. Requires root. Will impact all GPUs unless a single GPU is specified using the **-i** argument. This setting takes effect after the next reboot and is persistent.

-p, --reset-ecc-errors=TYPE

Reset the ECC error counters for the target GPUs. See the (*GPU ATTRIBUTES*) section for a description of ECC error counter types. Available arguments are 0|VOLATILE or 1|AGGREGATE. Requires root. Will impact all GPUs unless a single GPU is specified using the **-i** argument. The effect of this operation is immediate.

-c, --compute-mode=MODE

Set the compute mode for the target GPUs. See the (*GPU ATTRIBUTES*) section for a description of compute mode. Requires root. Will impact all GPUs unless a single GPU is specified using the **-i** argument. The effect of this operation is immediate. However, it does not persist across reboots. After each reboot compute mode will reset to "DEFAULT".

-dm, --driver-model**-fdm, --force-driver-model**

Enable or disable TCC driver model. For Windows only. Requires administrator privileges. `-dm` will fail if a display is attached, but `-fdm` will force the driver model to change. Will impact all GPUs unless a single GPU is specified using the `-i` argument. A reboot is required for the change to take place. See **Driver Model** for more information on Windows driver models.

-r, --gpu-reset

Trigger secondary bus reset of the GPU. Can be used to reset GPU HW state in situations that would otherwise require a machine reboot. Typically useful if a double bit ECC error has occurred. Requires `-i` switch to target specific device. Requires root. There can't be any applications using this particular device (e.g. CUDA application, graphics application like X server, monitoring application like other instance of `nvidia-smi`). There also can't be any compute applications running on any other GPU in the system. Only on supported devices from Fermi and Kepler family running on Linux.

GPU reset is not guaranteed to work in all cases. In some situations there may be HW components on the board that fail to revert back to an initial state following the reset request. This is more likely to be seen on Fermi-generation products vs. Kepler, and more likely to be seen if the reset is being performed on a hung GPU.

Following a reset, it is recommended that the health of the GPU be verified before further use. The `nvidia-healthmon` tool is a good choice for this test. If the GPU is not healthy a complete reset should be initiated by power cycling the node. `nvidia-healthmon` is distributed as a part of TDK <http://developer.nvidia.com/tesla-deployment-kit>

-ac, --applications-clocks=MEM_CLOCK,GRAPHICS_CLOCK

Specifies maximum <memory,graphics> clocks as a pair (e.g. 2000,800) that defines GPU's speed while running applications on a GPU. Only on supported device from Kepler family. Requires root.

-rac, --reset-application-clocks

Resets the application clocks to the default value. Only on supported device from Kepler family. Requires root.

-pl, --power-limit=POWER_LIMIT

Specifies maximum power limit in watts. Accepts integer and floating point numbers. Only on supported devices from Kepler family. Requires administrator privileges. Value needs to be between Min and Max Power Limit as reported by `nvidia-smi`.

[plus optional]**-i, --id=ID**

Modify a single specified GPU. The specified id may be the GPU/Unit's 0-based index in the natural enumeration returned by the driver, the GPU's board serial number, the GPU's UUID, or the GPU's PCI bus ID (as domain:bus:device.function in hex). It is recommended that users desiring consistency use either UUID or PCI bus ID, since device enumeration ordering is not guaranteed to be consistent between reboots and board serial number might be shared between multiple GPUs on the same board.

UNIT MODIFICATION OPTIONS**-t, --toggle-led=STATE**

Set the LED indicator state on the front and back of the unit to the specified color. See the (*UNIT ATTRIBUTES*) section for a description of the LED states. Allowed colors are 0|GREEN and 1|AMBER. Requires root.

[plus optional]**-i, --id=ID**

Modify a single specified Unit. The specified id is the Unit's 0-based index in the natural enumeration returned by the driver.

SHOW DTD OPTIONS**--dtd**

Display Device or Unit DTD.

[plus optional]**-f FILE, --filename=FILE**

Redirect query output to the specified file in place of the default stdout. The specified file will be overwritten.

-u, --unit

Display Unit DTD instead of device DTD.

GPU ATTRIBUTES

The following list describes all possible data returned by the **-q** device query option. Unless otherwise noted all numerical results are base 10 and unitless.

Timestamp

The current system timestamp at the time nvidia-smi was invoked. Format is "Day-of-week Month Day HH:MM:SS Year".

Driver Version

The version of the installed NVIDIA display driver. This is an alphanumeric string.

Attached GPUs

The number of accessible NVIDIA GPUs. Under Linux all NVIDIA GPUs are expected to be accessible.

Product Name

The official product name of the GPU. This is an alphanumeric string. For all products.

Display Mode

A flag that indicates whether a display is attached to the GPU. "Enabled" indicates an attached display. "Disabled" indicates otherwise.

Persistence Mode

A flag that indicates whether persistence mode is enabled for the GPU. Value is either "Enabled" or "Disabled". When persistence mode is enabled the NVIDIA driver remains loaded even when no active clients, such as X11 or nvidia-smi, exist. This minimizes the driver load latency associated with running dependent apps, such as CUDA programs. For all CUDA-capable products. Linux only.

Driver Model

On Windows, the TCC and WDDM driver models are supported. The driver model can be changed with the **(-dm)** or **(-fdm)** flags. The TCC driver model is optimized for compute applications. I.E. kernel launch times will be quicker with TCC. The WDDM driver model is designed for graphics applications

and is not recommended for compute applications. Linux does not support multiple driver models, and will always have the value of "N/A".

Current The driver model currently in use. Always "N/A" on Linux.

Pending The driver model that will be used on the next reboot. Always "N/A" on Linux.

Serial Number

This number matches the serial number physically printed on each board. It is a globally unique immutable alphanumeric value.

GPU UUID

This value is the globally unique immutable alphanumeric identifier of the GPU. It does not correspond to any physical label on the board.

VBIOS Version

The BIOS of the GPU board.

Inforom Version

Version numbers for each object in the GPU board's inforom storage. The inforom is a small, persistent store of configuration and state data for the GPU. All inforom version fields are numerical. It can be useful to know these version numbers because some GPU features are only available with inforoms of a certain version or higher.

If any of the fields below return Unknown Error additional Inforom verification check is performed and appropriate warning message is displayed.

Image Version Version for the OEM configuration data. Global version of the infoROM image. Image version just like VBIOS version uniquely describes the exact version of the infoROM flashed on the board in contrast to infoROM object version which is only an indicator of supported features.

OEM Object Version for the OEM configuration data.

ECC Object Version for the ECC recording data.

Power Object Version for the power management data.

PCI

Basic PCI info for the device. Some of this information may change whenever cards are added/removed/moved in a system. For all products.

Bus PCI bus number, in hex

Device PCI device number, in hex

Domain PCI domain number, in hex

Device Id PCI vendor device id, in hex

Sub System Id PCI Sub System id, in hex

Bus Id PCI bus id as "domain:bus:device.function", in hex

GPU Link information

The PCIe link generation and bus width

Current The current link generation and width. These may be reduced when the GPU is not in use.

Maximum The maximum link generation and width possible with this GPU and system configuration. For example, if the GPU supports a higher PCIe generation than the system supports then this reports the system PCIe generation.

Fan Speed

The fan speed value is the percent of maximum speed that the device's fan is currently intended to run at. It ranges from 0 to 100%. Note: The reported speed is the intended fan speed. If the fan is physically blocked and unable to spin, this output will not match the actual fan speed. Many parts do not report fan speeds because they rely on cooling via fans in the surrounding enclosure. For all discrete products with dedicated fans.

Performance State

The current performance state for the GPU. States range from P0 (maximum performance) to P12 (minimum performance).

Clocks Throttle Reasons

Retrieves information about factors that are reducing the frequency of clocks. Only on supported Tesla devices from Kepler family.

If all throttle reasons are returned as "Not Active" it means that clocks are running as high as possible.

Idle Nothing is running on the GPU and the clocks are dropping to Idle state. This limiter may be removed in a later release.

User Defined Clocks

GPU clocks are limited by user specified limit. E.g. set by `nvidia-smi --applications-clocks=`

SW Power Cap SW Power Scaling algorithm is reducing the clocks below requested clocks because the GPU is consuming too much power. E.g. SW power cap limit can be changed with `nvidia-smi --power-limit=`

HW Slowdown HW Slowdown (reducing the core clocks by a factor of 2 or more) is engaged.

This is an indicator of:

- * temperature being too high
- * External Power Brake Assertion is triggered (e.g. by the system power supply)
- * Power draw is too high and Fast Trigger protection is reducing the clocks
- * May be also reported during PState or clock change

****** This behavior may be removed in a later release

Unknown Some other unspecified factor is reducing the clocks.

Memory Usage

On-board memory information. Reported total memory is affected by ECC state. If ECC is enabled the total available memory is decreased by several percent, due to the requisite parity bits. The driver may also reserve a small amount of memory for internal use, even without active work on the GPU. For all products.

Total Total installed GPU memory.

Used Total memory allocated by active contexts.

Free Total free memory.

Compute Mode

The compute mode flag indicates whether individual or multiple compute applications may run on the GPU.

"DEFAULT" means multiple contexts are allowed per device.

"EXCLUSIVE_THREAD" means only one context is allowed per device, usable from one thread at a time.

"EXCLUSIVE_PROCESS" means only one context is allowed per device, usable from multiple threads at a time.

"PROHIBITED" means no contexts are allowed per device (no compute apps).

"EXCLUSIVE_PROCESS" was added in CUDA 4.0. Prior CUDA releases supported only one exclusive mode, which is equivalent to "EXCLUSIVE_THREAD" in CUDA 4.0 and beyond.

For all CUDA-capable products.

Utilization

Utilization rates report how busy each GPU is over time, and can be used to determine how much an application is using the GPUs in the system.

GPU Percent of time over the past second during which one or more kernels was executing on the GPU.

Memory Percent of time over the past second during which global (device) memory was being read or written.

Ecc Mode

A flag that indicates whether ECC support is enabled. May be either "Enabled" or "Disabled". Changes to ECC mode require a reboot. Requires Inforom ECC object version 1.0 or higher.

Current The ECC mode that the GPU is currently operating under.

Pending The ECC mode that the GPU will operate under after the next reboot.

ECC Errors

NVIDIA GPUs can provide error counts for various types of ECC errors. Some ECC errors are either single or double bit, where single bit errors are corrected and double bit errors are uncorrectable. Texture memory errors may be correctable via resend or uncorrectable if the resend fails. These errors are available across two timescales (volatile and aggregate). Single bit ECC errors are automatically corrected by the HW and do not result in data corruption. Double bit errors are detected but not corrected. Please see the ECC documents on the web for information on compute application behavior when double bit errors occur. Volatile error counters track the number of errors detected since the last driver load. Aggregate error counts persist indefinitely and thus act as a lifetime counter.

A note about volatile counts: On Windows this is once per boot. On Linux this can be more frequent. On Linux the driver unloads when no active clients exist. Hence, if persistence mode is enabled or there is always a driver client active (e.g. X11), then Linux also sees per-boot behavior. If not, volatile counts are reset each time a compute app is run.

Tesla and Quadro products from the Fermi and Kepler family can display total ECC error counts, as well as a breakdown of errors based on location on the chip. The locations are described below. Location-based data for aggregate error counts requires Inforom ECC object version 2.0. All other ECC counts require ECC object version 1.0.

Device Memory Errors detected in global device memory.

Register File Errors detected in register file memory.

L1 Cache Errors detected in the L1 cache.

L2 Cache Errors detected in the L2 cache.

Texture Memory Parity errors detected in texture memory.

Total Total errors detected across entire chip. Sum of **Device Memory**, **Register File**, **L1 Cache**, **L2 Cache** and **Texture Memory**.

Temperature

Readings from temperature sensors on the board. All readings are in degrees C. Not all products support all reading types. In particular, products in module form factors that rely on case fans or passive cooling do not usually provide temperature readings. See below for restrictions.

GPU Core GPU temperature. For all discrete and S-class products.

Power Readings

Power readings help to shed light on the current power usage of the GPU, and the factors that affect that usage. When power management is enabled the GPU limits power draw under load to fit within a predefined power envelope by manipulating the current performance state. See below for limits of availability.

Power State Power State is deprecated and has been renamed to Performance State in 2.285. To maintain XML compatibility, in XML format Performance State is listed in both places.

Power Management

A flag that indicates whether power management is enabled. Either "Supported" or "N/A". Requires Inforom PWR object version 3.0 or higher or Kepler device.

Power Draw

The last measured power draw for the entire board, in watts. Only available if power management is supported. This reading is accurate to within +/- 5 watts. Requires Inforom PWR object version 3.0 or higher or Kepler device.

Power Limit

The power management algorithm's power ceiling, in watts. Total board power draw is manipulated by the power management algorithm such that it stays under this value. Only available if power management is supported. Requires Inforom PWR object version 3.0 or higher or Kepler device. On Kepler devices Power Limit can be adjusted using `-pl,--power-limit=` switches.

Default Power Limit

The default power management algorithm's power ceiling, in watts. Power Limit will be set back to Default Power Limit after driver unload. Only on supported devices from Kepler family.

Min Power Limit

The minimum value in watts that power limit can be set to. Only on supported devices from Kepler family.

Max Power Limit

The maximum value in watts that power limit can be set to. Only on supported devices from Kepler family.

Clocks

Current frequency at which parts of the GPU are running. All readings are in MHz.

Graphics

Current frequency of graphics (shader) clock.

SM

Current frequency of SM (Streaming Multiprocessor) clock.

Memory

Current frequency of memory clock.

Applications Clocks

User specified frequency at which applications will be running at. Can be changed with `[-ac | --applications-clocks]` switches.

Graphics

User specified frequency of graphics (shader) clock.

Memory

User specified frequency of memory clock.

Default Applications Clocks

Default value of applications clocks. This are the applications clocks that will be used after system reboot or driver reload.

Graphics	Default value of applications clock of graphics (shader).
Memory	Default value of applications clock of memory clock.

Max Clocks

Maximum frequency at which parts of the GPU are design to run. All readings are in MHz.

Graphics	Maximum frequency of graphics (shader) clock.
SM	Maximum frequency of SM (Streaming Multiprocessor) clock.
Memory	Maximum frequency of memory clock.

Supported clocks

List of possible memory and graphics clocks combinations that the GPU can operate on (not taking into account HW brake reduced clocks). These are the only clock combinations that can be passed to `--applications-clocks` flag. Supported Clocks are listed only when `-q -d SUPPORTED_CLOCKS` switches are provided or in XML format.

Compute Processes

List of processes having compute context on the device.

Each Entry is of format "<pid>. <Process name>"

Used GPU Memory

Amount memory used on the device by the context. Not available on Windows when running in WDDM mode because Windows KMD manages all the memory not NVIDIA driver.

UNIT ATTRIBUTES

The following list describes all possible data returned by the `-q -u` unit query option. Unless otherwise noted all numerical results are base 10 and unitless.

Timestamp

The current system timestamp at the time `nvidia-smi` was invoked. Format is "Day-of-week Month Day HH:MM:SS Year".

Driver Version

The version of the installed NVIDIA display driver. Format is "Major-Number.Minor-Number".

HIC Info

Information about any Host Interface Cards (HIC) that are installed in the system.

Firmware Version

The version of the firmware running on the HIC.

Attached Units

The number of attached Units in the system.

Product Name

The official product name of the unit. This is an alphanumeric value. For all S-class products.

Product Id

The product identifier for the unit. This is an alphanumeric value of the form "part1-part2-part3". For all S-class products.

Product Serial

The immutable globally unique identifier for the unit. This is an alphanumeric value. For all S-class products.

Firmware Version

The version of the firmware running on the unit. Format is "Major-Number.Minor-Number". For all S-class products.

LED State

The LED indicator is used to flag systems with potential problems. An LED color of AMBER indicates an issue. For all S-class products.

Color The color of the LED indicator. Either "GREEN" or "AMBER".

Cause The reason for the current LED color. The cause may be listed as any combination of "Unknown", "Set to AMBER by host system", "Thermal sensor failure", "Fan failure" and "Temperature exceeds critical limit".

Temperature

Temperature readings for important components of the Unit. All readings are in degrees C. Not all readings may be available. For all S-class products.

Intake Air temperature at the unit intake.

Exhaust Air temperature at the unit exhaust point.

Board Air temperature across the unit board.

PSU

Readings for the unit power supply. For all S-class products.

State Operating state of the PSU. The power supply state can be any of the following: "Normal", "Abnormal", "High voltage", "Fan failure", "Heatsink temperature", "Current limit", "Voltage below UV alarm threshold", "Low-voltage", "I2C remote off command", "MOD_DISABLE input" or "Short pin transition".

Voltage PSU voltage setting, in volts.

Current PSU current draw, in amps.

Fan Info

Fan readings for the unit. A reading is provided for each fan, of which there can be many. For all S-class products.

State The state of the fan, either "NORMAL" or "FAILED".

Speed For a healthy fan, the fan's speed in RPM.

Attached GPUs

A list of PCI bus ids that correspond to each of the GPUs attached to the unit. The bus ids have the form "domain:bus:device.function", in hex. For all S-class products.

NOTES

On Linux, NVIDIA device files may be modified by nvidia-smi if run as root. Please see the relevant section of the driver README file.

The **-a** and **-g** arguments are now deprecated in favor of **-q** and **-i**, respectively. However, the old arguments still work for this release.

EXAMPLES

nvidia-smi -q

Query attributes for all GPUs once, and display in plain text to stdout.

nvidia-smi -q -d ECC,POWER -i 0 -l 10 -f out.log

Query ECC errors and power consumption for GPU 0 at a frequency of 10 seconds, indefinitely, and record to the file out.log.

nvidia-smi -c 1 -i GPU-b2f5f1b745e3d23d-65a3a26d-097db358-7303e0b6-149642ff3d219f8587cde3a8

Set the compute mode to "EXCLUSIVE_THREAD" for GPU with UUID "GPU-b2f5f1b745e3d23d-65a3a26d-097db358-7303e0b6-149642ff3d219f8587cde3a8".

nvidia-smi -q -u -x --dtd

Query attributes for all Units once, and display in XML format with embedded DTD to stdout.

nvidia-smi --dtd -u -f nvsmi_unit.dtd

Write the Unit DTD to nvsmi_unit.dtd.

nvidia-smi -q -d SUPPORTED_CLOCKS

Display supported clocks of all GPUs.

nvidia-smi -i 0 --applications-clocks 2500,745

Set applications clocks to 2500 MHz memory, and 745 MHz graphics.

Known Issues

- On Linux when X Server is running Used GPU Memory in Compute Processes section may contain value that is larger than the actual value. This will be fixed in a future release.
- On Linux GPU Reset may not successfully change pending ECC mode. A full reboot may be required to enable the mode change.

CHANGE LOG

=== Changes between nvidia-smi v3.295 and v4.304 ===

- * Reformatted non-verbose output due to user feedback. Removed pending information from table.
- * Print out helpful message if initialization fails due to kernel module not receiving interrupts
- * Better error handling when NVML shared library is not present in the system
- * Added new --applications-clocks switch
- * Added new filter to --display switch. Run with -d SUPPORTED_CLOCKS to list possible clocks on a GPU
- * When reporting free memory, calculate it from the rounded total and used memory so that values add up
- * Added reporting of power management limit constraints and default limit
- * Added new --power-limit switch
- * Added reporting of texture memory ECC errors
- * Added reporting of Clock Throttle Reasons

=== Changes between nvidia-smi v2.285 and v3.295 ===

- * Clearer error reporting for running commands (like changing compute mode)
- * When running commands on multiple GPUs at once N/A errors are treated as warnings.
- * nvidia-smi -i now also supports UUID
- * UUID format changed to match UUID standard and will report a different value.

=== Changes between nvidia-smi v2.0 and v2.285 ===

- * Report VBIOS version.
- * Added -d/--display flag to filter parts of data
- * Added reporting of PCI Sub System ID
- * Updated docs to indicate we support M2075 and C2075
- * Report HIC HWBC firmware version with -u switch
- * Report max(P0) clocks next to current clocks
- * Added --dtd flag to print the device or unit DTD
- * Added message when NVIDIA driver is not running
- * Added reporting of PCIe link generation (max and current), and link width (max and current).
- * Getting pending driver model works on non-admin
- * Added support for running nvidia-smi on Windows Guest accounts
- * Running nvidia-smi without -q command will output non verbose version of -q instead of help
- * Fixed parsing of -l/--loop= argument (default value, 0, to big value)
- * Changed format of pciBusId (to XXXX:XX:XX.X - this change was visible in 280)
- * Parsing of busId for -i command is less restrictive. You can pass 0:2:0.0 or 0000:02:00 and other variations
- * Changed versioning scheme to also include "driver version"

- * XML format always conforms to DTD, even when error conditions occur

- * Added support for single and double bit ECC events and XID errors (enabled by default with -l flag disabled for -x flag)

- * Added device reset -r --gpu-reset flags

- * Added listing of compute running processes

- * Renamed power state to performance state. Deprecated support exists in XML output only.

- * Updated DTD version number to 2.0 to match the updated XML output

SEE ALSO

On Linux, the driver README is installed as /usr/share/doc/NVIDIA_GLX-1.0/README.txt

AUTHOR

NVIDIA Corporation

COPYRIGHT

Copyright 2011-2012 NVIDIA Corporation.