



# **DengLin System Management Interface (dlsmi) V2 User Guide**

DL-DG/SW-038A-01

2023-04-30

Copyright©苏州登临科技有限公司，2019 - 2025，版权所有。

未经苏州登临科技有限公司事先书面同意，不得以任何形式或方式复制或传播本文件的任何部分。

## 商标和许可



和其它苏州登临科技有限公司的其它登临科技的图标为苏州登临科技有限公司的商标。本手册中提及的所有其他商标均为其各自所有者的财产。

## 通知

所购买的产品、服务和特性由苏州登临科技有限公司与客户签订的合同规定。本文件中描述的所有或部分产品、服务和特性可能不在采购范围或使用范围内。除非合同中另有规定，本文件中的所有声明、信息和建议均按“原样”提供，无任何明示或暗示的保证或陈述。

本手册中的信息如有更改，恕不另行通知。本文件在编制过程中已尽一切努力确保内容的准确性，本文件中的所有声明、信息和建议不构成任何明示或暗示的保证。

苏州登临科技有限公司

苏州工业园区扬富路11号南岸新地一期商务楼5号1101室，江苏，中国

<http://www.denglin.ai>

Email : support@denglin.ai

## Change History

Version	Change description
01	Initial version.

# Table of Contents

Table of Contents
Introduction
Synopsis
Options
GENERAL OPTIONS
LIST OPTIONS
SUMMARY OPTIONS
QUERY OPTIONS
DEVICE MODIFICATION OPTIONS
STATISTICS (EXPERIMENTAL)
SELECTIVE QUERY OPTIONS
MIG
Return Values
GPU Attributes
Timestamp
Driver Version
Attached GPUs
Product Name
Display Mode
Display Active
Persistence Mode
Accounting Mode
Accounting Buffer Size
Serial Number
Minor Number
VBIOS Version
MultiGPU Board
Inforom Version
PCI
Fan Speed
Performance State
Clocks Throttle Reasons
Compute Mode
Utilization
Temperature
Power Readings
Clocks
Application Clocks
Default Application Clocks
Max Clocks
Clock Policy
Supported Clocks
Processes
Note
Examples

## Introduction

DengLin System Management Interface (dlsmi) provides monitoring and management capabilities for the DengLin Goldwasser™ AI accelerator card. Metrics can be consumed directly by users via stdout, or provided by files via CSV and XML formats for scripting purposes.

Note that much of the functionality of dlsmi is provided by the underlying dML C-based library.

## Synopsis

```
dlsmi [OPTION1 [ARG1]] [OPTION2 [ARG2]] ...
```

## Options

### GENERAL OPTIONS

**-h, --help**

Print usage information and exit.

### LIST OPTIONS

**-L, --list-gpus**

Display a list of GPUs connected to the system.

**-B, --list-blacklist-gpus**

Display a list of blacklisted GPUs in the system.

### SUMMARY OPTIONS

**summary**

Show a summary of GPUs connected to the system.

[plus any of]

**-i, --id=**

Target a specific GPU.

**-f, --filename=**

Log to a specified file, rather than to stdout.

**-l, --loop=**

Probe until Ctrl+C at a specified second interval.

## QUERY OPTIONS

`-q`, `--query`

Display GPU information.

[plus any of]

`-i`, `--id=`

Target a specific GPU.

`-f`, `--filename=`

Log to a specified file, rather than to stdout.

`-x`, `--xml-format`

Produce XML output.

`--dtd`

When showing xml output, embed DTD.

`-d`, `--display=`

Display only selected information:

`MEMORY`, `UTILIZATION`, `ECC`, `TEMPERATURE`, `POWER`, `CLOCK`, `COMPUTE`, `PIDS`, `PERFORMANCE`,  
`SUPPORTED_CLOCKS`, `PAGE_RETIREMENT`, `ACCOUNTING`, `ENCODER_STATS`

Flags can be combined with comma, for example: `dlsmi -q -d ECC,POWER`.

Sampling data with max/min/avg is also returned for `POWER`, `UTILIZATION` and `CLOCK` display types.

Doesn't work with `-u` or `-x` flags.

`-l`, `--loop=`

Probe until Ctrl+C at a specified second interval.

`-lms`, `--loop-ms=`

Probe until Ctrl+C at a specified millisecond interval.

## DEVICE MODIFICATION OPTIONS

`-mig`, `--multi-instance-gpu=`

Enable or disable Multi Instance GPU: 0/DISABLED, 1/ENABLED

Requires root.

[plus optional]

`-i`, `--id=`

Target a specific GPU.

## STATISTICS (EXPERIMENTAL)

`stats`

Display device statistics.

`dlsmi stats -h` for more information.

## SELECTIVE QUERY OPTIONS

Allow the caller to pass an explicit list of properties to query.

[one of]

`--query-gpu=`

Information about GPU.

Call `--help-query-gpu` for more info.

`--query-supported-clocks=`

List of supported clocks.

Call `--help-query-supported-clocks` for more info.

`--query-compute-apps=`

List of currently active compute processes.

Call `--help-query-compute-apps` for more info.

[mandatory]

`--format=`

Comma separated list of format options:

`csv` - comma separated values (MANDATORY)

`noheader` - skip the first line with column headers

`nounits` - doesn't print units for numerical values

[plus any of]

`-i`, `--id=`

Target a specific GPU.

`-f`, `--filename=`

Log to a specified file, rather than to stdout.

`-l`, `--loop=`

Probe until Ctrl+C at a specified second interval.

`-lms`, `--loop-ms=`

Probe until Ctrl+C at a specified millisecond interval.

## MIG

`mig`

Provide controls for MIG management.

`dlsmi mig -h` for more information.

## Return Values

Return codes reflect whether the operation succeeded or failed.

- Return code 0 - Success
- Return code 1 - Failed
- Return code 2 - A supplied argument or flag is invalid

- Return code 3 - The requested operation is not available on target device
- Return code 4 - The current user does not have permission to access this device or perform this operation
- Return code 6 - A query to find an object was unsuccessful
- Return code 8 - A device's external power cables are not properly attached
- Return code 9 - DengLin driver is not loaded
- Return code 10 - DengLin Kernel detected an interrupt issue with a GPU
- Return code 12 - dIML Shared Library cannot be found or loaded
- Return code 13 - Local version of dIML does not implement this function
- Return code 14 - infoROM is corrupted
- Return code 15 - The GPU has fallen off the bus or has otherwise become inaccessible
- Return code 999 - Other error or internal driver error occurred

## GPU Attributes

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

### Timestamp

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

### Driver Version

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

### Attached GPUs

The number of DengLin GPUs in the system.

### Product Name

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

### Display Mode

A flag that indicates whether a physical display (for example, monitor) is currently connected to any of the GPU connectors. `Enabled` indicates an attached display. `disabled` indicates otherwise.

### Display Active

A flag that indicates whether a display is initialized on the GPU (for example, memory is allocated on the device for display). Display can be active even when no monitor is physically attached. `Enabled` indicates an active 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 DengLin driver remains loaded even when no active clients (such as X11 or dlsmi) exist. This minimizes the driver load latency associated with running dependent apps, such as CUDA programs.

## Accounting Mode

A flag that indicates whether accounting mode is enabled for the GPU.

When accounting is enabled, statistics are calculated for each compute process running on the GPU. Statistics can be queried during the lifetime or after termination of the process. The execution time of process is reported as 0 while the process is in running state and updated to actual execution time after the process has terminated.

## Accounting Buffer Size

Returns the size of the circular buffer that holds the list of processes that can be queried for accounting stats. This is the maximum number of processes that accounting information will be stored for, before information about oldest processes will get overwritten by information about new processes.

## Serial Number

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

## Minor Number

The minor number for the device is such that the DengLin device node file for each GPU will have the form `/dev/dri/renderD[minor number]`. Available only on Linux platform.

## VBIOS Version

The BIOS of the GPU board.

## MultiGPU Board

Whether or not this GPU is part of a multiGPU 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

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 information for the device. Some of this information may change whenever cards are added/removed/moved in a system.

- 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 Info  
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.
- Tx Throughput  
The GPU-centric transmission throughput across the PCIe bus in MB/s over the past 20 ms.
- Rx Throughput  
The GPU-centric receiving throughput across the PCIe bus in MB/s over the past 20 ms.

## 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%.

## 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.

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 the idle state. This limiter may be removed in a later release.

- Applications Clocks Setting

GPU clocks are limited by applications clocks setting. This setting can be changed using `dlsmi --applications-clocks=`.

- SW Power Cap

SW Power Scaling algorithm is reducing the clocks below requested clocks because the GPU is consuming too much power.

For example: SW power cap limit can be changed with `dlsmi --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 (for example, by the system power supply).
- Power draw is too high and Fast Trigger protection is reducing the clocks.
- Unknown some other unspecified factor is reducing the clocks.

## 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 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).

## 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 sample period, during which one or more kernels was executing on the GPU. The sample period may be between 1 second and 1/6 second depending on the product.

- Memory

Percent of time over the past sample period, during which global (device) memory was being read or written. The sample period may be between 1 second and 1/6 second depending on the product.

## Temperature

Readings from temperature sensors on the board. All readings are in degrees Celsius.

- GPU Shutdown Temp

The temperature at which a GPU will shut down.

- GPU Slowdown Temp

The temperature at which a GPU will begin slowing itself down in order to cool.

## Power Readings

Power readings clearly show 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 Management

A flag that indicates whether power management is enabled. Either "Supported" or "N/A".

- Power Draw

The last measured power draw for the entire board, in watts. Only available if power management is supported.

- Power Limit

The software power limit, in watts. Set by software such as dlsmi. Only available if power management is supported. Power Limit can be adjusted using `-pl,--power-limit= switches`.

- Enforced 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. This limit is the minimum of various limits such as the software limit listed above. Only available if power management is supported.

- 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.

- Min Power Limit

The minimum value in watts that power limit can be set to.

- Max Power Limit

The maximum value in watts that power limit can be set to.

## Clocks

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

- Core  
Current frequency of core (shader) clock.
- TE  
Current frequency of TE (Tensor Engine) clock.
- Memory  
Current frequency of memory clock.
- Video  
Current frequency of video (encoder + decoder + JPU) clocks.

## Application Clocks

User specified frequency at which applications will be running at.

## Default Application Clocks

Default frequency at which applications will be running at.

## Max Clocks

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

- Core  
Maximum frequency of core (shader) clock.
- TE  
Maximum frequency of TE (Tensor Engine) clock.
- Memory  
Maximum frequency of memory clock.
- Video  
Maximum frequency of video (encoder + decoder + JPU) clock.

## Clock Policy

User-specified settings for automated clocking changes such as auto boost.

- Auto Boost  
Indicates whether the auto boost mode is currently enabled for this GPU (On) or disabled for this GPU (Off). Shows N/A if boost is not supported. Auto boost allows dynamic GPU clocking based on power, thermal and utilization.
- Auto Boost Default  
Indicates the default setting for auto boost mode, either enabled (On) or disabled (Off). Shows N/A if boost is not supported.

## Supported Clocks

List of possible memory and coreclocks 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.

## Processes

List of processes having Compute Context on the device.

Each entry is of format `<GPU Index> <PID> <Process Name> <GPU Memory Usage> <JPU Util> <Decoder Util> <Encoder Util>`

`<GPU Index>` represents dIML Index of the device.

`<PID>` represents Process ID corresponding to the active Compute context.

`<Process Name>` represents process name for the Compute process.

`<GPU Memory Usage>` is amount of memory used on the device by the context.

Driver may reserve more DDR memory when channel DDR is allocated from processes.

`<JPU Util>` represent JPUs utilization.

`<Decoder Util>` represent decoders utilization, this is measured by session count.

`<Encoder Util>` represent encoders utilization, this is measured by session count.

## Note

In Linux, DengLin device files may be modified by dlsmi if you run it as root.

## Examples

- `dlsmi -q`  
Query attributes for all GPUs once, and display in plain text to stdout.
- `dlsmi -q -d POWER -i 0 -l 10 -f out.log`  
Query power consumption for GPU 0 at a frequency of 10 seconds, indefinitely, and record to file `out.log`.
- `dlsmi -mig 1`  
Enable GPU mig mode.
- `dlsmi mig -lgip`  
List mig mode gpu instance profiles.
- `dlsmi mig -cgi 0`  
Create gpu instance by using profile 0.