Intel® Vision Accelerator Design with the Intel® Movidius™ Myriad™ X VPU

HAL Configuration Guide

September 2019

Revision 1.3

September 2019
Document Number: 595850-1.3

Contents

Figures

Tables

Revision History

Date	Revision	Description
September 2019	1.3	Updated for OpenVINO 2019 R3 release.
April 2019	1.2	Added additional details. Reformatted for easier readability.
January 2019	1.1	For OpenVINO 2019 R1release.
September 2018	1.0	Gold release.
July 2018	0.8	Beta release.
April 2018	0.3	Initial release.

1.0 Introduction

This document provides information about configuring the high-density deep learning (HDDL) hardware abstract layer (HAL).

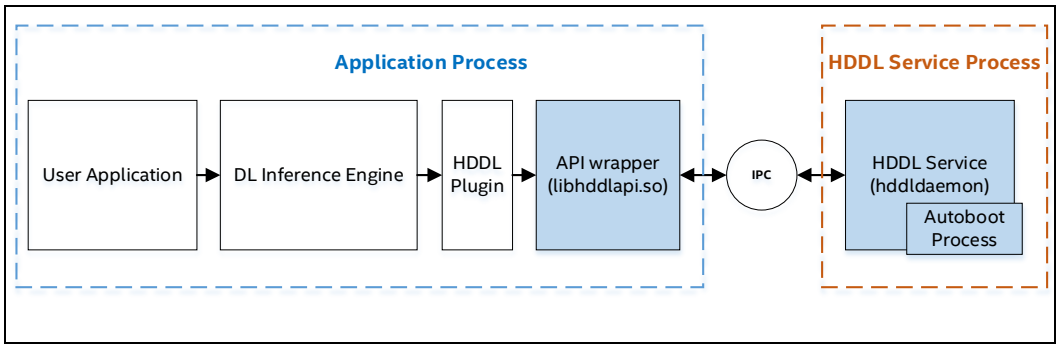
Note: Through the rest of this document the high-density deep learning hardware abstract layer is referred to as the "HAL" unless otherwise specified.

The HAL extracts the operations of the Intel® Vision Accelerator Design board that contains multiple Intel® Movidius™ Myriad™ X Vision Processing Units (VPUs). The HAL lets the Intel® Vision Accelerator Design board function as a Convolutional Neural Network (CNN) accelerator.

The Inference Engine API that is available in the Intel® Distribution of OpenVINO™ toolkit provides the HAL application interface, and an Inference Engine plugin (HDDLPlugin) access the HAL.

The HAL consists of two components: the HAL wrapper API library, and the high density deep learning daemon process. The wrapper API is called inside the Inference Engine plugin and communicates with the daemon.

Figure 1. HAL High-Level Architecture



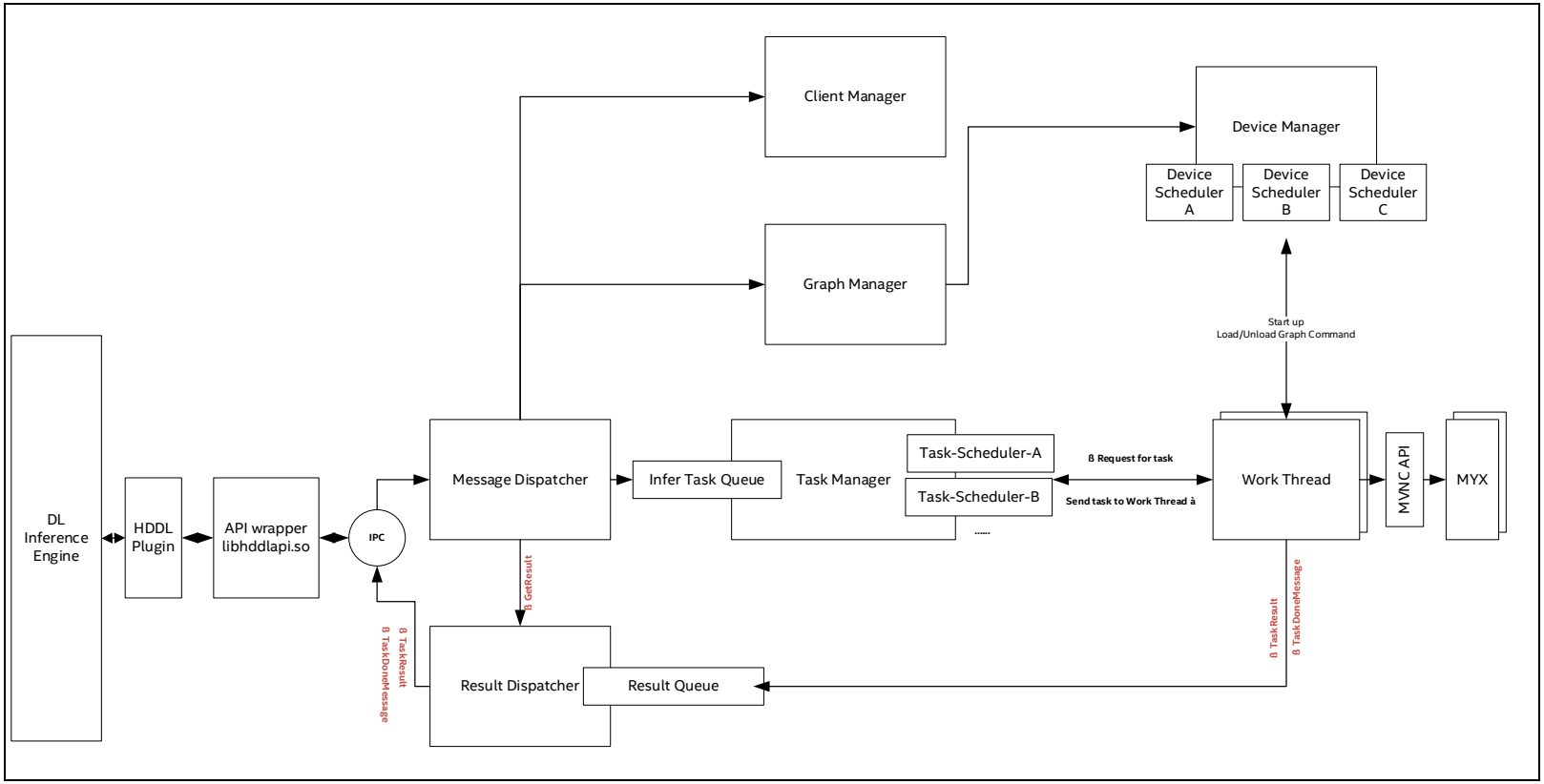


Figure 2. HAL Architecture

Table 1. HAL Component Features

Component	Features
Message Dispatcher	Dispatches messages and commands to corresponding the components.
Client Manager	Maintains the client status. Each Inference Engine instance creates one client with one graph inside the HAL.
Graph Manager	Maintain the graph status. Each Inference Engine graph creates one graph inside the HAL.
Task Manager	Inference messages and commands from a client are pushed to the pushed to the Task Manager, which caches all incoming inference requests. The Task Manager selects a specified task scheduler for each configuration file when booting and delivering inference tasks, according to the Task Scheduler. A client can create multiple tasks.
Device Manager	The Device Manager manages all Intel® Movidius™ Myriad™ X VPU devices and creates one work thread for each. Choose the correct Intel® Movidius™ Myriad™ X VPU to do inference tasks, according to the device scheduler that is loaded from configuration file at boot time.
Work Thread	Activated by the Device Manager. The work thread controlling one Intel® Movidius™ Myriad™ X VPU device. Key work thread tasks include: • Loading and unloading a graph to and from the Intel® Movidius™ Myriad™ X VPU device • Getting inference tasks from Task Manager and sending the tasks to the Intel® Movidius™ Myriad™ X VPU device • Getting inference result from Intel® Movidius™ Myriad™ X VPU device

The HAL uses configuration files to let you configure HAL behaviors. These configuration files are in `$HDDL_INSTALL_DIR/config/`. This guide provides information about the contents in each configuration file and the settings you can change.

Table 2. Configuration Files

Configuration File	Target	Configuration
`hddl_autoboot.config`	`autoboot`	• Autoboot settings • Firmware descriptions
`hddl_service.config`	`hddldaemon`	• Service feature settings • Fake device settings • Log level/Debugging settings • InfoPrinter settings for runtime statistics dumping
`hddl_api.config`	`libhddlapi.so`	• API settings • Log level settings

1.1 Terminology

Term	Description
CNN	Convolution Neural Network
HAL	Hardware Abstract Layer. In this document the HAL refers to the high-density deep learning hardware abstract layer.
Intel® Distribution of OpenVINO™ toolkit	Based on convolutional neural networks (CNN). The toolkit extends workloads across Intel® hardware and maximizes performance.

2.0 Autoboot Configuration

To change the Autoboot settings, modify the `hddl_autoboot.config` file.

When the `hddldaemon` starts, it launches the standalone Autoboot process. Autoboot is a complex two-stage process controlled by a configuration file that has the default filename `hddl_autoboot.config`.

The two-stage Autoboot process is:

1. Load firmware to the Intel® Movidius™ Myriad™ X VPU before the VPU boots.
2. Open the Intel® Movidius™ Myriad™ X VPU.

To simplify the device-managing logic in the `hddldaemon`, Autoboot helps load the firmware to the Intel® Movidius™ Myriad™ X VPU. This way, the `hddldaemon` doesn't handle the first VPU boot stage. Instead, the `hddldaemon` interfaces with the VPU after the firmware is loaded.

You have three ways to pass the `hddl_autoboot.config` file to the Autoboot process. Listed from highest to lowest use recommendation, these three ways are:

1. `$ HDDL_INSTALL_DIR/config/hddl_autoboot.config`
2. Linux: `$ HOME/.hddl_autoboot.config`
   Windows: `%HOMEPATH%\.hddl_autoboot.config`
3. `hddldaemon boot-config` option:
   `$ hddldaemon --boot-config user_autoboot.config`

The following table lists the `hddl_autoboot.config` configuration settings that are described in the sections that follow.

Autoboot Configuration File Setting	Function
Security Setting	Define which user group can access the HAL service.
Autoboot Settings	Define what should happen during the Autoboot process.
Unboot Device Settings	Specifies the VIDs and PIDs for each Intel® Movidius™ Myriad™ X VPU before they are booted.
Boot Device Setting	Specifies the VIDs and PIDs for each Intel® Movidius™ Myriad™ X VPU after they are booted.
Firmware Settings	Define an array of descriptions that Autoboot uses to identify and load firmware to Intel® Movidius™ Myriad™ X VPUs.

2.1 Security Setting

Only one security setting is available:

Security Setting	Function
`user_group`	Define which user group can access the HAL service.

Note: Default values are in bold, blue text.

2.1.1 `user_group`

Define which user group can access the HAL service. Only one user group can be provided.

Value	Action
`"users"`	This user group can access the HAL service.

2.2 Autoboot Settings

Define what should happen during the Autoboot process.

Autoboot Setting	Function
`work_mode`	Define how Autoboot should monitor Intel® Movidius™ Myriad™ X VPU activity.
`startup_wait_timeout`	Define how long the Intel® Movidius™ Myriad™ X VPUs should be ready before proceeding.
`abort_if_hw_reset_failed`	Define whether Autoboot should exit if an Intel® Movidius™ Myriad™ X VPU fails during a hardware reset.
`abort_if_device_num_not_met`	Define whether Autoboot should exit if an Intel® Movidius™ Myriad™ X VPU is not found.
`total_device_num`	Specify the number of Intel® Movidius™ Myriad™ X VPUs for which you want Autoboot to load firmware.

2.2.1 work_mode

Define how Autoboot should monitor Intel® Movidius™ Myriad™ X VPU activity

Value	Action
"hotplug"	Autoboot monitors Intel® Movidius™ Myriad™ X VPU devices with hotplug in/out events generated by them.
"scan"	**Default**. Autoboot occasionally scans the Intel® Movidius™ Myriad™ X VPU devices.

2.2.2 startup_wait_timeout

Time in milliseconds in which all Intel® Movidius™ Myriad™ X VPUs should be ready before proceeding to the next step. The ready state is indicated by the device being found and firmware is loaded to the device.

Value in Milliseconds	Action
A number of your choice	Wait this amount of time after all Intel® Movidius™ Myriad™ X VPU devices are ready before proceeding.
"100000"	**Default**. All Intel® Movidius™ Myriad™ X VPU devices are ready for **100000** milliseconds before proceeding.

2.2.3 abort_if_hw_reset_failed

Define whether Autoboot and the `hddldaemon` should exit if an Intel® Movidius™ Myriad™ X VPU fails.

Value	Action
"false"	Do not exit if an Intel® Movidius™ Myriad™ X VPU fails.
"true"	**Default**. Exit if an Intel® Movidius™ Myriad™ X VPU fails.

2.2.4 abort_if_device_num_not_met

Define whether Autoboot and the `hddldaemon` should exit if an Intel® Movidius™ Myriad™ X VPU is not found.

Value	Action
`"true"`	Exit Autoboot and the `hddldaemon` if: • No new Intel® Movidius™ Myriad™ X VPU device is detected within `startup_wait_timeout` milliseconds since the last Intel® Movidius™ Myriad™ X VPU was detected. • and the number of the detected Intel® Movidius™ Myriad™ X VPU devices are less than required.
`"false"`	**Default**. Do not exit Autoboot and the `hddldaemon`.

2.2.5 total_device_num

The number of Intel® Movidius™ Myriad™ X VPU devices for which you want Autoboot to load firmware.

Value	Action
"1" through "8" The number of VPU devices on the Intel® Vision Accelerator Design board.	Autoboot is to load firmware for the specified number of Intel® Movidius™ Myriad™ X VPU devices.
`"8"`	**Default**. Autoboot is to load firmware for eight Intel® Movidius™ Myriad™ X VPU devices.

Example scenario: Using Selected Intel® Movidius™ Myriad™ X VPUs

If you don't want the Intel® Vision Accelerator Design's board to control or use all the connected Intel® Movidius™ Myriad™ X VPUs, use the `"total_device_num"` setting in `hddl_autoboot.config` to set the number of VPUs to use. This example uses three VPUs:

```
"total_device_num": 3
```

2.3 Unboot Device Settings

Only one unboot device setting is available. This setting specifies VIDs and PIDs for each Intel® Movidius™ Myriad™ X VPU before they are booted.

Unboot Device Setting	Function
`unboot_device_settings`	Specify the VID and PID for each Intel® Movidius™ Myriad™ X VPU before they are booted.

2.3.1 `unboot_device_setting`

The VID and PID of unbooted Intel® Movidius™ Myriad™ X VPU devices.

Autoboot uses this information to count the number of unbooted Intel® Movidius™ Myriad™ X VPU device. This determines whether Autoboot needs to load firmware for idle devices.

Option	Value	Definition
VID	`"0x03e7"`	**Default**. The VID for each unbooted Intel® Movidius™ Myriad™ X VPU.
PID	`"0x2485"`	**Default**. The PID for each unbooted Intel® Movidius™ Myriad™ X VPU.

2.4 Boot Device Setting

Only one boot device setting is available. This setting specifies VIDs and PIDs for the Intel® Movidius™ Myriad™ VPU.

Boot Device Setting	Function
`boot_device_setting`	Specify the VID and PID for each Intel® Movidius™ Myriad™ X VPU after they are booted.

2.4.1 boot_device_setting

The VID and PID of booted Intel® Movidius™ Myriad™ X VPU devices.

Autoboot uses this information to count the number of booted Intel® Movidius™ Myriad™ X VPU to see if all required devices have booted.

Option	Value	Definition
VID	`"0x03e7"`	**Default**. The VID for each booted Intel® Movidius™ Myriad™ X VPU
PID	`"0x2485"`	**Default**. The PID for each booted Intel® Movidius™ Myriad™ X VPU

2.5 Firmware Settings

Firmware settings provide an array of descriptions that Autoboot uses to identify and load firmware to Intel® Movidius™ Myriad™ X VPUs.

Firmware Setting	Function
`name`	Specify the firmware filename.
`num`	Specify how many Intel® Movidius™ Myriad™ X VPUs are to be loaded with firmware.
`path`	Specify the firmware path
`vid` and `pid`	Define the VID and PID for each Intel® Movidius™ Myriad™ X VPU that is to be loaded with firmware.

2.5.1 name

The user-specified firmware name.

You can have only one `"default"` firmware.

Value	Action
Firmware file name of your choice	Autoboot is to use this user-defined firmware file.
`"default"`	**Default**. Autoboot is to use the default firmware file.

2.5.2 num

The number of Intel® Movidius™ Myriad™ X VPUs that are to be loaded with firmware. `num` is used with the firmware `name` to determine the specific devices on which firmware is loaded, and the name of the firmware file.

Value	Action
"0"	Do not load firmware on any of the Intel® Movidius™ Myriad™ X VPUs
"0" to "7"	Load firmware on the specified number of Intel® Movidius™ Myriad™ X VPUs
"8"	**Default**. Load firmware on all available Intel® Movidius™ Myriad™ X VPUs

2.5.3 path

The firmware location. This is where Autoboot is to look for the firmware.

Value	Action
User-defined path	Autoboot is to look in this directory for the firmware file.
/usr/local/lib/mvnc/MvNCAPI-ma2480.mvcmd	**Default**. Autoboot is to look in this directory for the firmware.

2.5.4 vid and pid

The USB firmware VID and PID. Autoboot uses the VID and PID to count the number of Intel® Movidius™ Myriad™ X VPU devices loaded with a specified firmware. The count determines whether the firmware is installed on all specified devices.

Option	Value	Definition
vid	**0x03e7**	**Default**. The VID for each booted Intel® Movidius™ Myriad™ X VPU
pid	**0xf63b**	**Default**. The PID for each booted Intel® Movidius™ Myriad™ X VPU

3.0 Service Configuration

To control the `hddldaemon` behavior, modify the `hddl_service.config` file in `HDDL_INSTALL_DIR/config/`

You have three ways to pass the `hddl_service.config` file to the `hddldaemon` process. Listed from highest to lowest recommendation, these three ways are:

- `$ HDDL_INSTALL_DIR/config/hddl_service.config`
- Linux: `$ HOME/.hddl_service.config`
  Windows: `% HOMEPATH%\.hddl_service.config`
- Create, specify and pass the configuration file to the HAL service from the command line. Use `-c` as an option.
  Example: `$ hddldaemon -c user_config.config`.

The following table lists the `hddl_service.config` configuration settings that are described in the following sections.

Service Configuration File Setting	Function
Service Settings	Configure the `hddldaemon` to manage things like graphs, tasks, and devices.
Security Settings	Define which user group can access the Intel® Vision Accelerator Design's board HAL service.
Device Settings	Define the VID/PID for the Intel® Movidius™ Myriad™ X VPU device to monitor its hot-plug incidents.
Scheduler Settings	Manage: • Tag Scheduler: The number of Intel® Movidius™ Myriad™ X VPU device that can process predictable tasks. • Stream Scheduler: • Bypass Scheduler: Control individual Intel® Movidius™ Myriad™ X VPUs. • Squeeze Scheduler: Manage a default scheduler.
Fake Device Settings	Used for evaluation only. This is for instances in which you do not yet have the necessary Intel® Vision Accelerator Design's board hardware.
Log Level Settings	Set the messages and instances in which to log information in a file.
Debug Settings	
Debug Process Setting	
Info Printer Setting	Define how and when to print information.

Example scenario: Check the Device Status

This scenario prints status information that you can control through the `hddl_service.config`, as described in the Log Level Setting file, Info Printer Setting.

This scenario is often used to check the status of each connected Intel® Movidius™ Myriad™ X VPU. For example, you might want to get the device ID, the network, or the network FPS. You can get this information with:

```
''debug_service'': true,
''device_snapshot_mode'': ''full'',
```

Your output is tables similar to those in device_snapshot_mode.

If, instead, you prefer a graph similar to that shown in graph_snapshot_style, use

```
''debug_service'': true,
''graph_snapshot_mode'': ''base'',
```

To learn the clients' status, such as how many clients are connected to the service, or the client FPS, use:

```
''client_snapshot_mode'': ''base'',
```

Your client is similar to that shown in client_snapshot_style.

3.1 Service Settings

The service settings configure `hddldaemon` features. These components are implemented inside the HAL to manage clients, graphs, tasks, and devices.

Service Setting	Function
`wait_seconds_before_exit`	Define how long the `hddldaemon` is to wait to exit after completing all tasks, or tell it not to exit.
`mvnc_log_level`	Specify how much information is to be included in the log file.
`common_timeout`	Set the timeout setting for common Intel® Movidius™ Neural Compute SDK (Intel® Movidius™ NCSDK) operations
`alloc_graph_timeout`	Set the timeout for allocating graph operation in the Intel® Movidius™ NCSDK
`max_cached_graph_number`	The maximum number of graphs that can be stored on an Intel® Movidius™ Myriad™ X VPU at the same time.
`update_timetaken_interval`	Define how often the Squeeze Scheduler is to collect the inference time for each graph.

Service Setting	Function
`task_scheduler`	Define the order in which tasks are to be handled.
`server_max_task_number`	Set the maximum number of tasks you can cash in the high-density HAS service.
`client_max_task_number`	Set the maximum number of tasks you can cash for each client in the high-density HAS service.

3.1.1 `wait_seconds_before_exit`

The `hddldaemon` waits `wait_seconds_before_exit` before exiting after all tasks are finished.

Value in in Seconds	Action
0 or greater	The `hddldaemon` exits after this number of seconds.
Less than 0	The `hddldaemon` does not exit.
`-1`	**Default**. The `hddldaemon` does not exit.

3.1.2 `mvnc_log_level`

Specify the log level control for the Intel® Movidius™ NCSDK API library. The higher the number, the less information in the log. The range is 0 ~ 4.

Value	Action
0	Log debug information, informational messages, warnings, errors, and fatal errors: `* MVLOG_DEBUG`
1	Log informational messages, warnings, errors, and fatal errors: `* MVLOG_INFO`
2	Log warnings, errors, and fatal errors: `* MVLOG_WARN`
`3`	**Default.** Log errors and fatal errors: `* MVLOG_ERROR`
4	Log only fatal errors: `* MVLOG_FATAL`

3.1.3 common_timeout

Set the timeout setting for common Intel® Movidius™ NCSDK operations, such as inference, get result, and others. If the operation doesn't finish within `common_timeout` seconds, a timeout error is returned and the HAL service on the Intel® Vision Accelerator Design board resets the Intel® Movidius™ Myriad™ VPU.

`common_timeout` includes the timeout settings for all other operations, except for open_device_timeout and `alloc_graph_timeout`.

Value	Action
1 - x	User-defined timeout
`1000`	**Default**. Set a 1000 second timeout

3.1.4 alloc_graph_timeout

`alloc_graph_timeout` is the timeout for allocating graph operation in Intel® Movidius™ NCSDK. If the operation doesn't finish within `alloc_graph_timeout`, a timeout error is returned and the high-density deep learning service resets the Intel® Movidius™ Myriad™ VPU.

Value	Action
`12000`	**Default**. Set a 12000 second timeout

3.1.5 max_cached_graph_number

`max_cached_graph_number` is the maximum number of graphs that can be cached on an Intel® Movidius™ Myriad™ VPU at the same time, where cached means storing, not running, the graphs.

This is beneficial for cases that require frequently loading and unloading graphs.

Value	Action
1 - x	The number of graphs that can be cached at a time.
`4`	**Default**. Cache up to four graphs on a single Intel® Movidius™ Myriad™ VPU at the same time.

3.1.6 update_timetaken_interval

Use `update_timetaken_interval` for Squeeze Device Scheduler settings. The Squeeze Device Scheduler collects the inference time for each graph at every `update_timetaken_interval` milliseconds. You can use this information in the Squeeze Device Scheduler.

Value in Milliseconds	Action
1 - x	Collect the inference time at this user-defined interval.
`1000`	**Default**. Collect the inference time every 1000 milliseconds.

3.1.7 task_scheduler

Task scheduling policy.

Value	Action
`fcfs`	First come, first serviced. Store all inference tasks for one graph from all clients in first in, first out order (FIFO). The task are processed in order, beginning with the first in.
`polling`	**Default**. Store all inference tasks for one graph from each client in an individual FIFO. Process the tasks in each FIFO in turn.

3.1.8 server_max_task_number

Sets the maximum number of cached tasks in the HAL service.

Value	Action
1 - x	Cache up to this user-defined number of tasks in the HAL service
`2000`	**Default**. Cache up to 2000 tasks in the HAL service

3.1.9 client_max_task_number

`client_max_task_number` is the maximum number of cached tasks for each client in the HAL service.

Value	Action
1 - x	Cache up to this user-defined number of client tasks in the HAL service
`0`	**Default**. Do not limit the number of cached client tasks.

3.2 Security Settings

Only one security setting is available.

Service Setting	Function
`user_group`	Define the user group that can access the HAL service.

3.2.1 `user_group`

`user_group` defines the user group that can access the HAL service. You can identify only one user group.

Value	Action
`users`	**Default**. Allow access to this user group.

3.3 Device Settings

Two device settings are available.

Service Setting	Function
`device_vid`	VID works with PID to identify Intel® Movidius™ Myriad™ X VPUs
`device_pid`	PID works with VID to identify Intel® Movidius™ Myriad™ X VPUs

3.3.1 `device_vid` and `device_pid`

`device_vid` and `device_pid` work together to identify individual Intel® Movidius™ Myriad™ X VPUs. The HAL service uses these IDs to monitor Intel® Movidius™ Myriad™ X VPU device hot-plug events.

Option	Value	Action
device_vid	`"0x03e7"`	Use this VID with the PID to refer to a specific Intel® Movidius™ Myriad™ X VPU.
device_pid	`"0xf63b"`	Use this PID with the VID to refer to a specific Intel® Movidius™ Myriad™ X VPU.

3.4 Scheduler Settings

The scheduler settings are different from the other options in this guide. Scheduler settings include four tag schedulers let you specialize how graphs are assigned to specific the Intel® Movidius™ Myriad™ X VPUs for processing.

The Scheduler Settings, including the Tag Schedulers are:

Service Setting	Function
Device Schedulers: The Tag, Stream, and Bypass Device Schedulers are for special circumstances. The Squeeze Scheduler is adequate for most circumstances.	
`graph_tag_map`	**Task Device Scheduler**: Provides task predictability by using a configuration map.
`stream_device_number`	**Stream Device Scheduler**: Processes a video stream in an uninterrupted context.
`bypass_device_number`	**Bypass Device Scheduler**: Control network deployment among Intel® Movidius™ Myriad™ X VPUs.
`device_schedule_interval`	**Squeeze Device Scheduler (Default)**: Manages scheduling the Intel® Movidius™ Myriad™ X VPUs across networks without user intervention.

Service Setting	Function
Additional Squeeze Scheduler Options	
max_cycle_switch_out	
max_task_number_switch_out	

If you have both stable and variable workloads, such as in surveillance scenarios, you might need frame-by-frame detection (stable FPS). At the same time, the classification workload depends on variable video content. In this case, it is best to compute the device number for the stable workload, and put the device number tag in the scheduler. Use the `graph_tag_map`.

For example, with a stable workload that requires three devices, and if other networks are available to use the `Squeeze Scheduler` for control, you can use this configuration:

```
''graph_tag_map'' {
    ''detect_network'': 3
},
```

If your network consists of variable workloads, do not change the scheduler settings. Instead, let the default Squeeze Scheduler balance the workload between all networks and Intel® Movidius™ Myriad™ X VPUs..

3.4.1 Subclass

Ignored. Set to `0`.

Value	Action
0	**Default**.

3.4.2 graph_tag_map

These settings are for the Tag Device Scheduler, which is used to process predictable tasks.

The Tag Device Scheduler manages multiple Intel® Movidius™ Myriad™ X VPUs. This scheduler accepts neural network graphs that are attached with a valid tag, and then allocates the tagged graphs to the VPUs, according to a configuration map. When the graph is no longer used, it is deallocated from the VPUs.

This scheduler provides a stable computing resource, which is necessary in an example such as the following surveillance example.

In surveillance, the camera IP address connected to an NVR (video gateway) system is usually fixed after deployment. To perform face detection on each frame in each video stream, the workload for face detection is calculable, using:

- The number of video streams
- The video stream frame rate
- The deep learning network used for "face detect"
- The processing capability of one Intel® Movidius™ Myriad™ X VPU for this network.

Use the configuration map to define how many devices that one graph, with a specified tag, can load. Pass the configuration map to the API.

For more information about the API, see the Intel® Distribution of OpenVINO toolkit API or the Intel® Movidius™ Myriad™ X HAL API User Guide.

Option	Value	Action
`"graph_tag_map" :`	Intel® Movidius™ Myriad™ X VPU to graph association. See the example below.	The VPU indicated processes the tagged graph that is associated with it.
`"graph_tag_map" :`	`{},`	**Default**. No associations.

Configuration map example:

```
"graph_tag_map" : {
     "AA":    3,
     "BB":    1,
},
```

With this configuration:

- Intel® Movidius™ Myriad™ X VPU #3 processes the graph tagged with `"AA"`
- Intel® Movidius™ Myriad™ X VPU #1 processes the graph tagged with `"BB"`

3.4.3 `stream_device_number`

These settings are for the Stream Device Scheduler.

With some context-aware networks, such as RNN and LSTM, successive frames in a video stream should be processed in an un-interrupted context. The Stream Scheduler meets this goal.

The Stream Device Scheduler manages Intel® Movidius™ Myriad™ X VPUs and only accepts networks that have an attached Stream ID. One Stream ID is allocated to one Intel® Movidius™ Myriad™ X VPU. All inference with the same Stream ID is directed to that VPU to do deep learning inference.

If two networks have the same stream ID, they are allocated to two VPUs. This is useful when one stream or frame must go through two deep learning processes.

If all Stream Device Scheduler managed VPUs have an allocated stream ID, then new networks that don't have a stream ID have no VPU and are rejected. When a network is no longer used, it is deallocated from Intel® Movidius™ Myriad™ X VPUs.

The Stream Scheduler adopts context awareness networks.

Option	Value	Action
`stream_device_number`	User-defined ID	Assign this Intel® Movidius™ Myriad™ X VPU with the user-defined ID.
`stream_device_number`	0	**Default**. Assign the Intel® Movidius™ Myriad™ X VPU with Device ID 0

3.4.4 bypass_device_number

These settings are for the Bypass Device Scheduler, which you can use to load and unload specified networks from Intel® Movidius™ Myriad™ X VPUs.

Sometimes, you might want to control individual Intel® Movidius™ Myriad™ X VPUs. For example, you might want to load or unload specific networks on specific VPUs according to a defined pattern. The Bypass Scheduler achieves this goal.

The Bypass Device Scheduler manages a user-defined number of Intel® Movidius™ Myriad™ X VPUs. You can load and unload networks from these VPUs individually.

The Bypass Device Scheduler defines the number of Intel® Movidius™ Myriad™ X VPU devices to assign to the Bypass Device Scheduler.

Value	Action
User-defined number	Assign this VPU number to the Bypass Device Scheduler.
0	**Default**.

Example Scenario: Better control of an individual Intel® Movidius™ Myriad™ X VPU

This example uses the `bypass_device_number` to achieve better control of Intel® Movidius™ Myriad™ X VPU number 5. The Bypass Scheduler can now control VPU number in the service, and you can use APIs to load and unload the network on this device.

```
"bypass_device_number": 5,
```

3.4.5 device_schedule_interval

These settings are for the Squeeze Device Scheduler. The Squeeze Device Scheduler is the default option.

With this option, you do not need to change the configuration file, and the Squeeze Device Scheduler manages the workload balance between all networks and Intel® Movidius™ Myriad™ X VPUs.

Deep learning tasks are usually unpredictable and you don't care how the neural networks are deployed between Intel® Movidius™ Myriad™ X VPUs. Applications offload neural networks to the VPUs, do deep learning inference to reach results, and the HAL system handles efficient scheduling and achieves high inference throughput.

If you don't specify a different scheduler for any or all of your an Intel® Movidius™ Myriad™ X VPUs, the Squeeze Device Scheduler manages all deep learning neural networks between VPUs.

The Squeeze Device Scheduler performs periodical scheduling to balance the workload among graphs and devices. While you do not need to change any settings, but if you want to, you can change the scheduling period.

Value in Milliseconds	Action
User-defined time	The user-defined length of scheduling time.
5000	**Default**.

3.4.6 max_cycle_switch_out and max_task_number_switch_out

These settings are for the Squeeze Device Scheduler. While you don't have to change any settings for the Squeeze Device Scheduler, you can use the `max_cycle_switch_out` and `max_task_number_switch_out` if you want to to define the circumstances under which graphs are swapped out of an Intel® Movidius™ Myriad™ X VPU.

When resources are constrained, the Squeeze Device Scheduler can swap graphs that have lower workloads out of an Intel® Movidius™ Myriad™ X VPU. The graph is swapped in:

- After the `max_cycle_switch_out` scheduling cycle, if the pending task number is less than or equal to `max_task_number_switch_out`.
- When the pending task number is greater than `max_task_number_switch_out`.

Option	Value	Action
`max_cycle_switch_out`	3	**Default**.

`max_task_number_swith_out`	`20`	**Default**.

3.5 Fake Device Settings

You can run the Intel® Vision Accelerator Design board;s service without the installing the hardware. To do so, use a software fake device as a dumb backend for integration. The fake returns a value and makes the system run. It doesn't load a graph or do inference.

Use a fake device when you want to evaluate the Compact R board HAL before you have the necessary hardware.

The fake device settings are:

Service Setting	Function
`enable_fake_device`	Turn on the fake device.
`fake_device_number`	Set the number of fake Intel® Movidius™ Myriad™ X VPUs to create.
`queue_capacity`	The number of tasks that can be cached in one fake device.
`wait_timeout`	The wait timeout in milliseconds when calling the `loadTensor` or `getResult` methods.
`alloc_graph_latency`	The average latency in milliseconds for allocating a graph on a fake device.
`load_tensor_latency`	The average latency in milliseconds of loading tensor on fake device.
`get_result_latency`	The average latency in milliseconds to get results on a fake device.
`latency_distribution_stddev`	The standard deviation of normal distribution the latency fake device obeys. Units in millisecond.

Example fake device settings

```
"enable_fake_device": true,
"fake_device_number": 5,
```

3.5.1 `enable_fake_device`

Use fake device instead of Intel® Movidius™ Myriad™ X VPU in the Compact R board HAL Service, if the value is set as true.

Value	Action
`"true"`	Use a fake device.
`"false"`	**Default**. Use an Intel® Movidius™ Myriad™ X VPU, not a fake device.

3.5.2 fake_device_number

The number of fake Intel® Movidius™ Myriad™ X VPUs to create.

Value	Action
User-defined number	Create this number of fake device.
`3`	**Default**. Create three fake devices

3.5.3 queue_capacity

The number of tasks that can be cached in one fake device.

Value	Action
User-defined number	Cache this number of tasks in one fake device.
`10`	**Default**. Cache up to 10 tasks in one fake device.

3.5.4 wait_timeout

The wait timeout in milliseconds when calling the `loadTensor` or `getResult` methods.

Value in milliseconds	Action
User-defined number	Wait this amount of time.
`5000`	**Default**. Wait 5000 seconds.

3.5.5 alloc_graph_latency

The average latency in milliseconds for allocating a graph on a fake device.

Value in Milliseconds	Action
User-defined number	Use this latency.
`5`	**Default**. Use a 5-millisecond latency.

3.5.6 load_tensor_latency

The average latency in milliseconds of loading tensor on fake device.

Value in Milliseconds	Action
User-defined number	Use this latency.
`5`	**Default**. Use a 5-millisecond latency.

3.5.7 get_result_latency

The average latency in milliseconds to get results on a fake device.

Value in Milliseconds	Action
User-defined number	Use this latency.
`10`	**Default**. Use a 10-millisecond latency

3.5.8 latency_distribution_stddev

The standard deviation of normal distribution the latency fake device obeys.

Value	Action
User-defined number	Use this standard deviation.
`1.0`	**Default**. Use a standard deviation of 1.0

3.6 Log Level Setting

These settings allow users to get different level log messages to understand the runtime status of HAL service.

Log Level Setting	Function
`log_frequent`	Define whether to log messages frequently or infrequently.
`log_debug`	Define whether to log debug messages.
`log_process`	Define whether to log process messages.
`log_info`	Define whether to log informational messages.
`log_warn`	Define whether to log warning messages.
`log_error`	Define whether to log error messages.
`log_fatal`	Define whether to log fatal error messages.

3.6.1 `log_frequent`

Define whether to log messages frequently or infrequently.

Value	Action
"on"	Log messages frequently.
"off"	**Default**. Log messages infrequently.

3.6.2 `log_debug`

Define whether to log debug messages.

Value	Action
"on"	Log debug messages.
"off"	**Default**. Do not log debug messages.

3.6.3 log_process

Define whether to log process messages. This is the primary Debug Process Setting configuration, and it controls the work process log of every component in the HAL service.

Value	Action
"on"	Log process messages.
"off"	**Default**. Do not log process messages.

3.6.4 log_info

Define whether to log informational messages.

Value	Action
"off"	Do not log informational messages.
"on"	**Default**. Log informational message.

3.6.5 log_warn

Define whether to log warning messages. Print log with log level = warn.

Value	Action
"off"	Do not log warning messages.
"on"	**Default**.

3.6.6 log_error

Define whether to log error messages.

Value	Action
"off"	Do not log error messages.
"on"	**Default**.

3.6.7 log_fatal

Whether print log with log level = fatal.

Value	Action
"off"	Do not log fatal messages.
"on"	**Default**. Log fatal messages

3.7 Debug Settings

Debug Setting	Function
debug_service	Define whether to display service log information.
graph_identity	

3.7.1 debug_service

Display service log information if set as true. Master switch of all *log_xxx* messages.

Value	Action
"false"	Do not display service log information.
"true"	**Default**. Display service log information.

3.7.2 graph_identity

IDs to distinguish one graph from another. When there are not enough graphs, you can choose "name" to create multiple graphs for testing.

Value	Action
"name"	Define an ID for multiple graphs
"hash"	**Default**.

3.8 Debug Process Settings

The Debut Process settings lists the key components in service. If you encounter a problem, turn on logging for all or some components, depending on whether you know where the problem happened. For help turning on these settings, see Log Level Setting.

Before you can enable any configurations in this section, set `log_process` to "`true`".

Debug Setting	Function
`message_dispatcher`	Choose whether to provide message dispatcher-related debug information.
`client_manager`	Choose whether to provide client manager-related debug information.
`graph_manager`	Choose whether to provide graph manager-related debug information.
`task_manager`	Choose whether to provide task-manager-related debug information.
`task_scheduler`	Choose whether to provide task scheduler-related debug information.
`device_manager`	Choose whether to provide device manager-related debug information.
`device_scheduler`	Choose whether to provide device scheduler-related debug information.
`device`	Choose whether to provide device-related debug information.
`device_work_thread`	Choose whether to provide device work thread-related debug information.
`result_dispatcher`	Choose whether to provide result dispatcher-related debug information.
`change_operation`	Choose whether to provide change operation-related debug information.
`graph_mapper`	Choose whether to provide graph mapper-related debug information.
`info_printer`	Choose whether to provide printer-related debug information.

3.8.1 `message_dispatcher`

Choose whether to provide message dispatcher-related debug information.

Value	Action
"`on`"	Display message dispatcher related debug information
`"off"`	**Default**. Do not display message dispatcher related debug information

3.8.2 `client_manager`

Choose whether to provide client manager-related debug information.

Value	Action
"`on`"	Display client manager-related debug information.

`"off"`	**Default**. Do not display client manager-related debug information.

3.8.3 graph_manager

Choose whether to provide graph manager-related debug information.

Value	Action
"on"	Display graph manager-related debug information.
`"off"`	**Default**. Do not display graph manager-related debug information.

3.8.4 task_manager

Choose whether to provide task manager-related debug.

Value	Action
"on"	Display graph task-related debug information.
`"off"`	**Default**. Do not display task manager-related debug information.

3.8.5 task_scheduler

Choose whether to provide task scheduler-related debug information.

Value	Action
"on"	Display task scheduler-related debug information
`"off"`	**Default**. Do not display task scheduler-related debug information.

3.8.6 device_manager

Choose whether to provide device manager-related debug information.

Value	Action
"on"	Display device manager-related debug information
`"off"`	**Default**. Do not display device manager-related debug information.

3.8.7 device_scheduler

Choose whether to provide device scheduler-related debug information.

Value	Action
"on"	Display device scheduler-related debug information.
"off"	**Default**. Do not display device scheduler-related debug information.

3.8.8 device

Choose whether to provide device-related debug information.

Value	Action
"on"	Display device-related debug information.
"off"	**Default**. Do not display device-related debug information.

3.8.9 device_work_thread

Choose whether to provide device work thread-related debug information.

Value	Action
"on"	Display device work thread-related debug information.
"off"	**Default**. Do not display device work thread-related debug information.

3.8.10 result_dispatcher

Choose whether to provide result dispatcher-related debug information.

Value	Action
"on"	Display result dispatcher-related debug information.
"off"	**Default**. Do not display device result dispatcher-related debug information.

3.8.11 change_operation

Choose whether to provide change operation-related debug information.

Value	Action
"on"	Display change operation-related debug information.
"off"	**Default**. Do not display change operation-related debug information.

3.8.12 graph_mapper

Choose whether to provide graph mapper-related debug information.

Value	Action
"on"	Display graph mapper-related debug information.
"off"	**Default**. Do not display graph mapper-related debug information.

3.8.13 info_printer

Choose whether to provide info printer-related debug information.

Value	Action
"on"	Display info printer-related debug information.
"off"	**Default**. Do not display infor printer-related debug information

3.9 Info Printer Setting

Info Printer is a high-density deep learning service component that collects information like the FPS of each different component, device utilization, device manager snapshot, task manager snapshot, and so on. The collected information helps you understand the status and running mode of the HAL Service.

Note: You must enable `log_info` before you can use these settings.

Info Printer Setting	Function
`enable`	Determine whether to enable the information printer.
`print_interval`	Set the print interval in milliseconds.
`client_fps`	Determine whether to print each client frame rate.
`device_fps`	Print the device working state.
`service_fps`	Determine whether to print the HAL service's serving rate.
`graph_fps`	Determine whether to print each graph's process rate.
`device_utilization`	Determine whether to print each device's service utilization.
`memory_usage`	Determine whether to print the service's memory usage.
`device_snapshot_mode`	Set the display mode for device snapshot.
`device_snapshot_style`	Set the style for device snapshot.
`client_snapshot_mode`	Set the display mode for client snapshots.
`client_snapshot_style`	Set the style for client snapshots.
`graph_snapshot_mode`	Set the display mode for graph snapshots.
`graph_snapshot_style`	Set the display style for a graph snapshots.
`task_snapshot_mode`	Set the display mode for task snapshots.
`task_snapshot_style`	Set the display style for task snapshots.

3.9.1 `enable`

Determine whether to enable the information printer.

Value	Action
`"false"`	Disable the printer.
`"true"`	**Default**. Enable the printer.

3.9.2 `print_interval`

Set the print interval in milliseconds.

Value	Action
user-defined interval	Print in this interval.
`"5000"`	**Default**. Print every 5000 miliseconds.

3.9.3 client_fps

Determine whether to print each client frame rate. The frame rate is defined as the rate of frame input to the HAL service from each client. The task may not complete.

Value	Action
"on"	Print the frame rate for the client.
"off"	**Default**. Do not print the client frame rate.

3.9.4 device_fps

Determine whether to print each device's working rate. States the task execution speed on each Intel® Movidius™ Myriad™ X VPU device.

Value	Action
"on"	Print the device working state.
"off"	**Default**. Do not print the device's working state.

3.9.5 service_fps

Determine whether to print the HAL service's serving rate. The serving rate is the FPS sum from all Intel® Movidius™ Myriad™ X VPU devices.

Value	Action
"on"	Print the serving rate.
"off"	**Default**. Do not print the serving rate.

3.9.6 graph_fps

Determine whether to print each graph's process rate. The process rate includes `input_fps` and `output_fps`.

Value	Action
"on"	Print each graph's process rate.
"off"	**Default**.

3.9.7 device_utilization

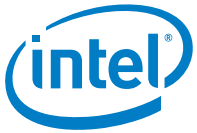
Determine whether to print each device's service utilization. “Device utilization = Idle time / unit interval”. Idle time is the time on inference task in device.

Value	Action
“on”	Print each device’s service utilization.
"off"	**Default**. Do not print each device's service utilization

3.9.8 memory_usage

Determine whether to print the service's memory usage.

Value	Action
“on”	Print the print service’s memory usage.
"off"	**Default**. Do not print the service’s memory useage.



3.9.9 device_snapshot_mode

Set the display mode for device snapshot.

Value	Action
"base"	
"full"	
"none"	**Default**. Snapshot is disabled.

Base Mode

```
[21:34:33.0918][2566]I[DeviceManager.cpp:721] DeviceSnapshot(RegularPrint):
| deviceId   | 6                | 7                | 5                | 4                | 3                | 2                | 1                | 0                |
| device     | 5.2-ma2480       | 5.1-ma2480       | 3.1-ma2480       | 3.2-ma2480       | 7.1-ma2480       | 9.2-ma2480       | 7.2-ma2480       | 9.1-ma2480       |
| thermal    | 56.58(0)         | 56.37(0)         | 54.21(0)         | 54.42(0)         | 53.33(0)         | 52.90(0)         | 55.29(0)         | 52.90(0)         |
+------------+------------------+------------------+------------------+------------------+------------------+------------------+------------------+------------------+
| status     | RUNNING          | RUNNING          | RUNNING          | RUNNING          | RUNNING          | RUNNING          | RUNNING          | RUNNING          |
| util%      | 71.90 %          | 71.90 %          | 71.90 %          | 61.20 %          | 60.31 %          | 72.77 %          | 72.78 %          | 72.76 %          |
| fps        | 108.53           | 109.09           | 109.09           | 46.12            | 46.80            | 109.18           | 108.87           | 109.18           |
| curGraph   | google.graph     | google.graph     | google.graph     | google3.graph    | google2.graph    | google5.graph    | google5.graph    | google5.graph    |
+------------+------------------+------------------+------------------+------------------+------------------+------------------+------------------+------------------+
| status     | WAIT_GRAPH       | WAIT_GRAPH       | WAIT_GRAPH       | WAIT_GRAPH       | WAIT_GRAPH       | WAIT_GRAPH       | WAIT_GRAPH       | WAIT_GRAPH       |
| util%      |                  |                  |                  |                  |                  |                  |                  |                  |
| fps        |                  |                  |                  |                  |                  |                  |                  |                  |
| curGraph   |                  |                  |                  |                  |                  |                  |                  |                  |
+------------+------------------+------------------+------------------+------------------+------------------+------------------+------------------+------------------+
```

Full Mode

```
[21:31:42.7171][1562]I[DeviceManager.cpp:721] DeviceSnapshot(RegularPrint):
| deviceId   | 7                 | 6                 | 4                 | 5                 | 2                 | 0                 | 3                 | 1                 |
| device     | 5.1-ma2480        | 5.2-ma2480        | 3.1-ma2480        | 3.2-ma2480        | 7.1-ma2480        | 9.1-ma2480        | 9.2-ma2480        | 7.2-ma2480        |
| thermal    | 60.63(0)          | 60.21(0)          | 59.36(0)          | 56.80(0)          | 55.51(0)          | 55.29(0)          | 56.37(0)          | 54.86(0)          |
+------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| status     | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           |
| util%      | 100.00 %          | 100.00 %          | 100.00 %          | 84.46 %           | 84.72 %           | 61.06 %           | 61.04 %           | 61.08 %           |
| fps        | 108.80            | 108.34            | 108.93            | 46.33             | 46.53             | 109.17            | 108.51            | 109.13            |
| curGraph   | google.graph      | google.graph      | google.graph      | google2.graph     | google3.graph     | google5.graph     | google5.graph     | google5.graph     |
| loadTime   | 20180706 21:30:59 | 20180706 21:30:59 | 20180706 21:30:59 | 20180706 21:31:19 | 20180706 21:31:33 | 20180706 21:31:39 | 20180706 21:31:39 | 20180706 21:31:39 |
| runTime    | 00:00:43          | 00:00:43          | 00:00:43          | 00:00:22          | 00:00:09          | 00:00:03          | 00:00:03          | 00:00:03          |
| inference  | 4698              | 4689              | 4698              | 1081              | 429               | 331               | 329               | 331               |
| prevGraph  |                   |                   |                   |                   | google3.graph     | google5.graph     | google5.graph     | google5.graph     |
| loadTime   |                   |                   |                   |                   | 20180706 21:31:26 | 20180706 21:31:15 | 20180706 21:31:15 | 20180706 21:31:15 |
| unloadTime |                   |                   |                   |                   | 20180706 21:31:31 | 20180706 21:31:38 | 20180706 21:31:38 | 20180706 21:31:38 |
| runTime    |                   |                   |                   |                   | 00:00:05          | 00:00:23          | 00:00:23          | 00:00:23          |
| inference  |                   |                   |                   |                   | 254               | 2572              | 2569              | 2572              |
+------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| status     | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        |
| util%      |                   |                   |                   |                   |                   |                   |                   |                   |
| fps        |                   |                   |                   |                   |                   |                   |                   |                   |
| curGraph   |                   |                   |                   |                   |                   |                   |                   |                   |
| loadTime   |                   |                   |                   |                   |                   |                   |                   |                   |
| runTime    |                   |                   |                   |                   |                   |                   |                   |                   |
| inference  |                   |                   |                   |                   |                   |                   |                   |                   |
| prevGraph  |                   |                   |                   |                   |                   |                   |                   |                   |
| loadTime   |                   |                   |                   |                   |                   |                   |                   |                   |
| unloadTime |                   |                   |                   |                   |                   |                   |                   |                   |
| runTime    |                   |                   |                   |                   |                   |                   |                   |                   |
| inference  |                   |                   |                   |                   |                   |                   |                   |                   |
+------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
```

3.9.10 device_snapshot_style

Set the style for device snapshot.

Value	Action
"tape"	
"table"	**Default**.

Figure 3. Example device_snapshot_style Tape Style

```
[21:33:43.7698][2062]I[DeviceManager.cpp:721] DeviceSnapshot(RegularPrint):
|   deviceId: 5              |   deviceId: 7              |   deviceId: 6              |   deviceId: 4              |   deviceId: 3              |   deviceId: 2              |   deviceId: 1              |   deviceId: 0              |
|     device: 7.1-ma2480     |     device: 3.2-ma2480     |     device: 9.2-ma2480     |     device: 3.1-ma2480     |     device: 9.1-ma2480     |     device: 5.1-ma2480     |     device: 7.2-ma2480     |     device: 5.2-ma2480     |
|    thermal: 57.23(0)       |    thermal: 57.44(0)       |    thermal: 56.80(0)       |    thermal: 55.51(0)       |    thermal: 54.42(0)       |    thermal: 60.21(0)       |    thermal: 57.23(0)       |    thermal: 59.57(0)       |
+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+
|     status: RUNNING        |     status: RUNNING        |     status: RUNNING        |     status: RUNNING        |     status: RUNNING        |     status: RUNNING        |     status: RUNNING        |     status: RUNNING        |
|      util%: 100.00 %       |      util%: 100.00 %       |      util%: 100.00 %       |      util%: 83.48 %        |      util%: 83.08 %        |      util%: 100.00 %       |      util%: 100.00 %       |      util%: 100.00 %       |
|   curGraph: google.graph   |   curGraph: google.graph   |   curGraph: google.graph   |   curGraph: google2.graph  |   curGraph: google3.graph  |   curGraph: google5.graph  |   curGraph: google5.graph  |   curGraph: google5.graph  |
|       fps: 108.70          |       fps: 108.40          |       fps: 108.67          |       fps: 45.35           |       fps: 45.15           |       fps: 108.65          |       fps: 108.74          |       fps: 108.62          |
|   loadTime: 20180706 21:33:33 |   loadTime: 20180706 21:33:33 |   loadTime: 20180706 21:33:33 |   loadTime: 20180706 21:33:27 |   loadTime: 20180706 21:33:31 |   loadTime: 20180706 21:33:24 |   loadTime: 20180706 21:33:24 |   loadTime: 20180706 21:33:24 |
|    runTime: 00:00:10       |    runTime: 00:00:10       |    runTime: 00:00:10       |    runTime: 00:00:15       |    runTime: 00:00:11       |    runTime: 00:00:18       |    runTime: 00:00:19       |    runTime: 00:00:18       |
| inference#: 1128           | inference#: 1125           | inference#: 1128           | inference#: 723            | inference#: 536            | inference#: 2057           | inference#: 2060           | inference#: 2054           |
|  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |
|   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |
| unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                |
|    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |
| inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                |
+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+
|     status: WAIT_GRAPH     |     status: WAIT_GRAPH     |     status: WAIT_GRAPH     |     status: WAIT_GRAPH     |     status: WAIT_GRAPH     |     status: WAIT_GRAPH     |     status: WAIT_GRAPH     |     status: WAIT_GRAPH     |
|      util%:                |      util%:                |      util%:                |      util%:                |      util%:                |      util%:                |      util%:                |      util%:                |
|   curGraph:                |   curGraph:                |   curGraph:                |   curGraph:                |   curGraph:                |   curGraph:                |   curGraph:                |   curGraph:                |
|       fps:                 |       fps:                 |       fps:                 |       fps:                 |       fps:                 |       fps:                 |       fps:                 |       fps:                 |
|   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |
|    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |
| inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                |
|  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |  prevGraph:                |
|   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |   loadTime:                |
| unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                | unloadTime:                |
|    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |    runTime:                |
| inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                | inference#:                |
+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+----------------------------+
```

Figure 4. Example `device_snapshot_style` Table Style

```
[21:31:42.7171][1562]I[DeviceManager.cpp:721] DeviceSnapshot(RegularPrint):
| deviceId    | 7                 | 6                 | 4                 | 5                 | 2                 | 0                 | 3                 | 1                 |
| device      | 5.1-ma2480        | 5.2-ma2480        | 3.1-ma2480        | 3.2-ma2480        | 7.1-ma2480        | 9.1-ma2480        | 9.2-ma2480        | 7.2-ma2480        |
| thermal     | 60.63(0)          | 60.21(0)          | 59.36(0)          | 56.80(0)          | 55.51(0)          | 55.29(0)          | 56.37(0)          | 54.86(0)          |
+-------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| status      | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           | RUNNING           |
| util%       | 100.00 %          | 100.00 %          | 100.00 %          | 84.46 %           | 84.72 %           | 61.06 %           | 61.04 %           | 61.08 %           |
| fps         | 108.80            | 108.34            | 108.93            | 46.33             | 46.53             | 109.17            | 108.51            | 109.13            |
| curGraph    | google.graph      | google.graph      | google.graph      | google2.graph     | google3.graph     | google5.graph     | google5.graph     | google5.graph     |
| loadTime    | 20180706 21:30:59 | 20180706 21:30:59 | 20180706 21:30:59 | 20180706 21:31:19 | 20180706 21:31:33 | 20180706 21:31:39 | 20180706 21:31:39 | 20180706 21:31:39 |
| runTime     | 00:00:43          | 00:00:43          | 00:00:43          | 00:00:22          | 00:00:09          | 00:00:03          | 00:00:03          | 00:00:03          |
| inference   | 4698              | 4689              | 4698              | 1081              | 429               | 331               | 329               | 331               |
| prevGraph   |                   |                   |                   |                   | google3.graph     | google5.graph     | google5.graph     | google5.graph     |
| loadTime    |                   |                   |                   |                   | 20180706 21:31:26 | 20180706 21:31:15 | 20180706 21:31:15 | 20180706 21:31:15 |
| unloadTime  |                   |                   |                   |                   | 20180706 21:31:31 | 20180706 21:31:38 | 20180706 21:31:38 | 20180706 21:31:38 |
| runTime     |                   |                   |                   |                   | 00:00:05          | 00:00:23          | 00:00:23          | 00:00:23          |
| inference   |                   |                   |                   |                   | 254               | 2572              | 2569              | 2572              |
+-------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
| status      | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        | WAIT_GRAPH        |
| util%       |                   |                   |                   |                   |                   |                   |                   |                   |
| fps         |                   |                   |                   |                   |                   |                   |                   |                   |
| curGraph    |                   |                   |                   |                   |                   |                   |                   |                   |
| loadTime    |                   |                   |                   |                   |                   |                   |                   |                   |
| runTime     |                   |                   |                   |                   |                   |                   |                   |                   |
| inference   |                   |                   |                   |                   |                   |                   |                   |                   |
| prevGraph   |                   |                   |                   |                   |                   |                   |                   |                   |
| loadTime    |                   |                   |                   |                   |                   |                   |                   |                   |
| unloadTime  |                   |                   |                   |                   |                   |                   |                   |                   |
| runTime     |                   |                   |                   |                   |                   |                   |                   |                   |
| inference   |                   |                   |                   |                   |                   |                   |                   |                   |
+-------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
```

3.9.11 client_snapshot_mode

Set the display mode for client snapshots.

Value	Action
"base"	
"table"	**Default**.

3.9.12 client_snapshot_style

Set the style for client snapshots.

Value	Action
"base"	
"table"	**Default**.

Figure 5. Example client_snapshot_style Table Style

```
[17:53:43.2117][6132]I[ClientManager.cpp:406] ClientSnapshot(RegularPrint):
| clientId     | 1                | 2                | 3                |
| clientName   | client1          | client5          | client4          |
| clientFps    | 42.93            | 28.02            | 42.75            |
| inferTimes   | 0                | 0                | 0                |
| waitNum      | 0                | 0                | 0                |
+--------------+------------------+------------------+------------------+
[17:53:43.2121][6132]I[GraphManager.cpp:698] GraphSnapshot(RegularPrint):
| graphId      | 284              | 282              | 288              |
| graphName    | graph9           | graph15          | graph12          |
| fpsIn        | 42.94            | 42.63            | 42.75            |
| fpsOut       | 42.94            | 42.63            | 42.75            |
| inferTimes   | 446              | 527              | 429              |
| waitTaskNum  | 0                | 0                | 0                |
| avgInferTime | 18.22            | 18.22            | 18.23            |
| devReqNum    | 0.39             | 0.39             | 0.39             |
| deviceNum    | 1                | 1                | 1                |
| graphTag     |                  |                  |                  |
| streamId     |                  |                  |                  |
+--------------+------------------+------------------+------------------+
```

3.9.13 graph_snapshot_mode

Set the display mode for graph snapshots.

Value	Action
"base"	
"none"	**Default**.

3.9.14 graph_snapshot_style

Set the display style for a graph snapshots.

Value	Action
"base"	
"table"	**Default**.

Figure 6. Example graph_snapshot_style Table Style

```
[21:32:00.7361][1562]I[GraphManager.cpp:641] GraphSnapshot(RegularPrint):
| graphId      | 1                | 6                | 3                | 5                |
| graphName    | google.graph     | google5.graph    | google2.graph    | google3.graph    |
| fpsIn        | 325.47           | 325.33           | 45.16            | 45.21            |
| fpsOut       | 325.76           | 325.47           | 45.16            | 45.21            |
| nWaitTasks   | 129              | 130              | 0                | 0                |
| avgInferTime | 18.10            | 18.23            | 16.70            | 16.68            |
| devReqNum    | 4.11             | 4.15             | 0.38             | 0.38             |
+--------------+------------------+------------------+------------------+------------------+
```

3.9.15 task_snapshot_mode

Set the display mode for task snapshots.

Value	Action
"base"	
"none"	**Default**.

3.9.16 task_snapshot_style

Set the display style for task snapshots.

Value	Action
"list"	**Default**.

4.0 API Configuration

The Inference Engine API in the Intel® Distribution of OpenVINO™ provides the API, and an Inference Engine plugin (`HDDLPlugin`) accesses the HAL.

This chapter provides instructions to modify the API configuration file, `hddl_api.config`, to control the `libhddlapi.so` behavior.

You have two ways to pass the `hddl_api.config` file to the `hddldaemon`. Listed from highest to lowest use recommendation, these two ways are:

- `$ HDDL_INSTALL_DIR hddl_api.config`
- Linux: `$ HOME/.hddl_api.config`
  Windows: `%HOMEPATH%\.hddl_api.config`

4.1 API Settings

Two API settings are available:

API Setting	Function
`max_cache_task`	Limit the number of tasks that a client can send to service before it gets the first response.
`timeout`	Defines the length of time to wait for a response to each message sent to the high-density deep learning service.

4.1.1 `max_cache_task`

Limit the number of tasks that a client can send to service before it gets the first response.

Value	Action
User-defined number	
`"150"`	**Default**.

4.1.2 `timeout`

The Intel® Vision Accelerator Design board uses a client/service architecture in which the client and server communicate through messages.

This setting defines the length of time to wait for a response to each message sent to the high-density deep learning service.

Time in Seconds	Action
User-defined time	Wait this long for a response to message.
`"60"`	**Default**. Wait 60 seconds for a response to messages.

4.2 Log Level Settings

Log Level Setting	Function
`log_frequent`	Determine how often to print to the log file
`log_debug`	Determine whether to print debug messages
`log_process`	Determine whether to print process information
`log_info`	Determine whether to print informational messages
`log_warn`	Determine whether to print warning information
`log_error`	Determine whether to print non-fatal error information
`log_fatal`	Determine whether to print fatal error information

The high-density deep learning service and API provide detailed log printing options for the debug levels described in this section. Use the log level settings in this section in the `hddl_service.config` file to control the logged information.

The `Debug Process Setting` lists the key components in service. If you encounter a problem, turn on logging for all or some components, depending on whether you know where the problem happened.

Note: Before you can enable the Debug Process Settings, you must set `log_process` to "on".

```
''log_process'': ''on'',
''message_dispatcher'': ''on'',
……
''info_printer'': ''on''
```

4.2.1 log_frequent

Determine how often to print to the log file.

Value	Action
"true"	Print to the log file frequently
`"false"`	**Default**. Print to the log file infrequently

4.2.2 log_debug

Determine whether to print debug messages: log level = debug.

Value	Action
"true"	Print debut messages
"false"	**Default**. Don't print debut messages

4.2.3 log_process

Determine whether to print process messages: log level = process.

Value	Action
"true"	**Print process messages**
"false"	**Default**. Don't print process messages

4.2.4 log_info

Determine whether to print log messages: log level = info.

Value	Action
"true"	Print log messages
"true"	**Default**. Don't print log messages

4.2.5 log_warn

Determine whether to print warning messages: log level = warn.

Value	Action
"false"	Don't print warning messages
"true"	**Default**. Print warning messages

4.2.6 log_error

Determine whether to print error messages: log level = error.

Value	Action
"false"	Don't print error messages
"true"	**Default**. Print error messages

4.2.7 log_fatal

Determine whether to print fatal messages: log level = fatal.

Value	Action
"false"	Don't print fatal messages
"true"	**Default**. Print fatal messages