

# HDDL-R Setup Guide

This guide is composed of two parts:

1. Installing the required IEI BSL reset driver
2. Configuration settings for the hddldaemon service.

NOTE: This guide does not apply to Uzel cards.

NOTENOTE: The upcoming revision of the IEI board does not need the reset driver (though they'll continue to sell current inventory). We need to figure out how to identify the different boards, and update the doc to address this.

## IEI HDDL-R Reset Driver Install Guide

If using an IEI HDDL-R board, such as the Mustang V100-MX8, an additional driver must be downloaded and installed for the card to work.

Use the start page to check for new drivers or search for software and documentation:

[IEI Downloads Home Page](#) (Search for: Mustang-V100-MX8)

Use these links for direct downloads of the latest driver, as of 8/13/2019:

[IEI Reset Driver Download, Linux](#)

[IEI Reset Driver Download, Windows](#)

With either driver, decompress the downloaded archive, enter the newly created directory, and run the install script:

On **Linux**:

- Run the install.sh script with sudo, or as root.

On **Windows**, do one of the following:

- **GUI**: Double-click install.bat
- **CLI**: Open a console with administrator privileges, cd into the directory, and run install.bat

## HDDL-R Service Configuration Guide

The hddldaemon is a system service, a binary executable that is run to manage the computational workload on the HDDL-R board. It's a required abstraction layer that handles inference, graphics processing, and any type of computation that should be run on the VPUs. Depending on the board configuration, there will be 8 or 16 VPUs (Myriad processing chips) on the board.

NOTE: graphics and other specialized computation may require some custom development.

The following lists a few of the possible configuration options. See this [non-existent reference document](#) for complete details. [DAVID] Want to make the workload document public as well.]

## Conventions Used in this Document

<IE> refers to the OpenVINO Inference Engine Directory (defaults follow):

Linux	/opt/intel/openvino/inference_engine
Windows	C:\Program Files (x86)\IntelSWTools\openvino\inference_engine

## Configuration File Location

<IE>\external\hddl\config\hddl\_service.config

## Service Configuration File Settings

NOTE: After changing a configuration file, the HDDL service (if running) must be restarted.

### Settings of Interest:

#### **device\_snapshot\_mode**

Changes the output of the hddl service to display a table with individual VPU statistics.

Default Setting
"device_snapshot_mode": "none"

Suggested Setting
"device_snapshot_mode": "full"

### Supported Settings

None (Default)  
Base  
Full

#### **device\_snapshot\_style**

Default Setting
"device_snapshot_style": "table"

Recommended Setting
"device_snapshot_style": "table"

The “table” setting is cleaner and presents labels on the left for each column. The “tape” setting prints the labels in each column.

### Supported Settings

Tape  
Table (default)

### user\_group

Restricts the service to group members.

Recommended      TBD for individual use cases (user or service)

Default Setting
“user_group”: “users”

The HDDL-R service may be restricted to a privileged group of users. The appropriate group will vary according to the local system configuration.

### Supported Settings

Valid groups on the current system. The “users” group is a default group that exists on Windows and most Linux distributions.

### Optional Recommended Settings

[DAVID] I think we want to provide details for the following:

“device\_fps”: “off”

~~This reports the FPS of each VPU, if set, this statistic will be added to each VPU’s column. The reported FPS seems to be much lower than “real life”, by a factor of ~6.~~ Need more info.

“device\_utilization”: “off”

Displays the percent of time each VPU is in use. Appears in the “table” column when turned on, or if “device\_fps” is turned on.

“memory\_usage”: “off”

Reports the amount of memory being used by each VPU.

“max\_cycle\_switchout”: 3

Requires the squeeze scheduler. Might speed up performance significantly, depending on the app.  
Explanation in HAL Config doc is gibberish (explanation of internal source code variables that are only understood by the devs). Could use a BKM or short guide on trying out different values.

Note: works in conjunction with: max\_task\_number\_switch\_out.

[DAVID] No opinion on these. Exclude?:

"client\_fps" : "off"

Reports the total FPS for the dispatching hddl\_service (which will have one or more clients per app).

### debug\_service

"debug\_service": "false" (default: "true")

[David] Not sure if we want to mention this, but the Linux version prints a lot of pointless debug information.

## HDDL-R Programming Reference

This reference is to educate a user on how to distribute a model across all 8 VPUs (video processing units) to maximize performance.

### Programming a C++ Application for the HDDL-R

#### Declare a Structure to Track Requests

The struct should hold:

1. a pointer to an inference request
2. an ID to keep track of the request.

```
struct Request {  
    InferenceEngine::InferRequest::Ptr inferRequest;  
    int frameidx;  
};
```

#### Declare a Vector of Requests:

```
// numRequests is the number of frames (max size, equal to number of VPUs in use)  
vector<Request> request(numRequests);
```

Declare and initialize 2 mutex variables:

- 1.) For each request
- 2.) For when all 8 requests are done.

#### Declare a Conditional Variable (indicates when at most 8 requests are done at a time)

For inference requests, use the asynchronous IE API calls:

```
// initialize infer request pointer – consult IE API for more detail  
request[i].inferRequest = executable_network.CreateInferRequestPtr();
```

```
// Run inference  
request[i].inferRequest->StartAsync();
```

### **Create a Lambda Function (for the parsing and display of results)**

Inside the Lambda body use the completion callback function:

```
request[i].inferRequest->SetCompletionCallback(InferenceEngine::InferRequest::Ptr context)
```

## **Programming a Python Application for the HDDL-R**

[DAVID] TBD will add after receiving source from Aleks Sutic.