# Introduction

This document contains a listing of possible values that can be used in an AppOptions.json file for the CVG HMD Operator View application.

To find the AppOptions.json file, see the root directory where the application executable is.

# Root Entries

These are the elements that go directly into the root json object.

VERSON

|  |  |
| --- | --- |
| Type | string |
| Description | Semantic versioning of the file format. While not currently used, this can be used to give the application context on the AppOptions contents if the file needs to start supporting different versions of the json file.  Note that this versioning data is used to track the file version, not the application version. The value of the version will be orthogonal to the application version. |
| Values | The value will be in the format “#.#.#”, where # is an integer value. The format will follow the semantic versioning format, where:   * The leftmost number is the major version. * The middle number is the minor version * The right number is the patch number.   What exactly these means is no fully defined, as versioning is planned to be used as-needed where useful. |
| Notes | -- |

\_\_comment\_\_

|  |  |
| --- | --- |
| Type | string |
| Description | Miscellaneous comment to anyone reading the AppOptions.json file in a text editor. |
| Values | Human readable text, with instructions on how the reader/developer can follow-up to get more technical information on the file format. |
| Notes | The \*.json file format does not support comments. |

\_debug\_ui

|  |  |
| --- | --- |
| Type | bool |
| Description | If true, initialize the application to show debug visualization.  This is not for any arbitrary debug data in the application, but specifically for debugging visual content such as the GUI. |
| Values | true to draw debug information.  false to not draw debug information (default). |
| Notes | -- |

\_fullscreen

|  |  |
| --- | --- |
| Type | bool |
| Description | In certain cases when debugging and troubleshooting, it may be convenient to not start the application in the expected full screen view. |
| Values | true to show the application full screen (default).  false to show the application windowed. |
| Notes | -- |

\_mousepad\_scale

|  |  |
| --- | --- |
| Type | float |
| Description | The scale of the mousepad graphic. |
| Values | The scale to draw the mousepad graphic. Where a larger scale value will result in a render larger than a rendering from a smaller number. |
| Notes | The value should be a number greater than 0. The scaling value is arbitrary and may require iteration to discover the best value |

\_mousepad\_x

|  |  |
| --- | --- |
| Type | Number |
| Description | Horizontal pixel offset value of the mousepad graphic. |
| Values | A value of 0 will be the horizontal center of the application viewport. |
| Notes | Screen coordinate system.  The value is in pixel units. |

\_mousepad\_y

|  |  |
| --- | --- |
| Type | number |
| Description | Vertical pixel offset value of the mousepad graphic. |
| Values | A value of 0 will be the vertical center of the application viewport. |
| Notes | Screen coordinate system.  The value is in pixel units. |

\_viewport\_width

|  |  |
| --- | --- |
| Type | number |
| Description | Horizontal pixel dimension of the video feed viewport. |
| Values | This is the width of the rectangle where streaming video feeds are shown. |
| Notes | The value is in pixel units. |

composite\_height

|  |  |
| --- | --- |
| Type | int |
| Description | Vertical pixel dimension of the video saved out for the composite video feed. |
| Values | This is the height of the video of the saved video for the composite video. |
| Notes | The value is in pixel units. |

composite\_width

|  |  |
| --- | --- |
| Type | int |
| Description | Horizontal pixel dimension of the video saved out for the composite video feed. |
| Values | This is the width of the video of the saved video for the composite video. |
| Notes | The value is in pixel units. |

\_viewport\_height

|  |  |
| --- | --- |
| Type | number |
| Description | Vertical pixel dimension of the video feed viewport. |
| Values | This is the height of the rectangle where streaming video feeds are shown. |
| Notes | The value is in pixel units. |

\_viewport\_offsx

|  |  |
| --- | --- |
| Type | number |
| Description | Horizontal pixel offset of the video feed viewport. |
| Values | This is the horizontal offset of the video feed viewport. |
| Notes | Screen coordinate system.  The value is in pixel units. |

\_viewport\_offsy

|  |  |
| --- | --- |
| Type | Number |
| Description | Vertical pixel offset of the video feed viewport. |
| Values | This is the vertical offset of the video feed viewport. |
| Notes | Screen coordinate system.  The value is in pixel units. |

carousel\_study

|  |  |
| --- | --- |
| Type | Array of carousel\_options entries. |
| Description | This will be the arbitrary list of entries shown in the study carousel. |
| Values | For the specification of the types of objects that should be in this array, see the section on *carousel Entries*.  Or, this can be a string to a json file containing a carousel\_system. |
| Notes | -- |

carousel\_series

|  |  |
| --- | --- |
| Type | Array of carousel\_options entries. |
| Description | This will be the arbitrary list of entries shown in the series carousel. |
| Values | For the specification of the types of objects that should be in this array, see the section on *carousel Entries*.  Or, this can be a string to a json file containing a carousel\_system. |
| Notes | -- |

carousel\_orientation

|  |  |
| --- | --- |
| Type | Array of carousel\_options entries. |
| Description | This will be the arbitrary list of entries shown in the orientation carousel. |
| Values | For the specification of the types of objects that should be in this array, see the section on *carousel Entries*.  Or, this can be a string to a json file containing a carousel\_system. |
| Notes | -- |

feed\_options

|  |  |
| --- | --- |
| Type | Array of feed\_options entries. |
| Description |  |
| Values | For the specification of the types of objects that should be in this array, see the section on *feed\_option Entries*. |
| Notes | Each *feed\_options* entry represents a separate video stream, from a camera. The system is currently hardcoded to 2 cameras, so *feed\_options* should have 2 entries. |

# carousel\_system Entries

These are external JSON files that contain carousel entries. This is so carousel definitions can be kept separate from a specific AppOption – as well as referenced between multiple AppOptions.

## Entries

entry

|  |  |
| --- | --- |
| Type | Array of carousel\_options entries. |
| Description | This will be the arbitrary list of entries shown in a carousel. |
| Values | For the specification of the types of objects that should be in this array, see the section on *carousel Entries*. |
| Notes | -- |

# carousel Entries

These are the members of the objects that populate the root *carousel* array.

## Entries

caption

|  |  |
| --- | --- |
| Type | string |
| Description | A longform text description. This string will be watermarked on image snapshots related to images taken in this snapshot stage. |
| Values | For the specification of the types of objects that should be in this array, see the section on *feed\_option Entries*. |
| Notes | -- |

icon\_filepath

|  |  |
| --- | --- |
| Type | string |
| Description | The icon to load for the carousel. This should be a path to a PNG. |
| Values | A file path to a png image.  The file path should either be a global path, or local from the application’s directory. |
| Notes | A recommended size for the icon is 256x256 |

id

|  |  |
| --- | --- |
| Type | string |
| Description | A string identifier for the entry.  This is for a currently unused feature – and either all *id*s in the entry array should be arbitrary unique values, or all *id*s should be empty strings. |
| Values | For the specification of the types of objects that should be in this array, see the section on *feed\_option Entries*. |
| Notes | -- |

label

|  |  |
| --- | --- |
| Type | string |
| Description | The short text shown next to the icon in the carousel. |
| Values | For the specification of the types of objects that should be in this array, see the section on *feed\_option Entries*. |
| Notes | It is suggested to use the convention of a short 1-word string, no longer than 4 characters – that are all capital letters.  This short string value will also be part of snapshot filenames. |

# feed\_options Entries

These are the members of the objects that populate the root *feed\_options* array.

They describe the various options for video feeds that can be streamed and drawn to the video viewport.

## Entries

default\_poll

|  |  |
| --- | --- |
| Type | string |
| Description | The polling method of video stream to use if a linux or windows override isn’t specified. |
| Values | A string from a set of expected values:   * **“cvusb”**   + The device index used by the OpenCV API. * **“devpath”**   + The linux device path. * **“external”**   + Run a program that can pipe its data to the application. * **“mmal”**   + Use the low-level RaspberryPi VideoCore/MMAL systems. * **“static”**   + Unused – previously brought in a static image that emulated a static video feed. * **“url”**   + A URL of a streaming video server. |
| Notes | This value is only used if a platform-specific override isn’t specified. For more information on these overrides, see windows\_poll and linux\_poll. |

windows\_poll

Similar to *default\_poll* but is an override specifically for Windows. The use of this value is optional.

linux\_poll

Similar to *default\_poll* but is an override specifically for Linux. The use of this value is optional.

dev\_path

|  |  |
| --- | --- |
| Type | string |
| Description | Used when the polling method (e.g., *default\_poll*) is set to “devpath” or “mmal”.  This will be used to specify a camera or video stream from a device path. |
| Values | The linux device path to poll the video stream from. |
| Notes | Has not been recently tested. |

index

|  |  |
| --- | --- |
| Type | int |
| Description | Used when polling method (e.g., *default\_poll*) is set to “cvusb”.  This will be used to specify a camera id from OpenCV, as the device to stream from. This OpenCV index is expected to address a USB device. |
| Values | A device index, usable by the OpenCV api. |
| Notes | Has not been recently tested. |

mmal\_ad\_gain

|  |  |
| --- | --- |
| Type | Array<float> |
| Description | The explicit static values for the analog and digital gain of the camera. |
| Values | An array of floats. Delete the entry or leave it as an empty array to have the camera use the default setting of dynamic gain adjustment.  To use, give the array two float values, the first value will be the analog gain, the second will be the digital gain. |
| Notes | Only supported for MMAL polling.  To use explicit values, the array must have exactly two entries. |

mmal\_index

|  |  |
| --- | --- |
| Type | int |
| Description | Used when polling method (e.g., *default\_poll*) is set to “mmal”.  This will be used to specify a camera id from MMAL, as the CSI port to stream from. |
| Values | A device index, referring to the CSI port to stream video via MMAL. |
| Notes | Which ports these values refer to will depend on the circuit board used – refer to board manufacturer documentation. |

mmal\_spam\_gain

|  |  |
| --- | --- |
| Type | bool |
| Description | Parameter for mmal values to output control properties whenever they’re changed. |
| Values | If true, then whenever a camera control property is changed, it will output the camera controls to the output terminal.  Note that this will work when explicit gains are set, but you will not see any changes for the locked gain values.  Also, the output does not happen at any expected rate. They are only output when the camera changes values, which it will do at arbitrary times. |
| Notes | Only supported for MMAL polling.  This may cause a lot of terminal spam. |

mmal\_wb\_gain

|  |  |
| --- | --- |
| Type | Array<float> |
| Description | The explicit static values for the white balance gain settings of the camera. |
| Values | An array of floats. Delete the entry or leave it as an empty array to have the camera use the default setting of dynamic white balance gain adjustment.  To use, give the array two float values, the first value will be the red gain, the second will be the blue gain. |
| Notes | Only supported for MMAL polling.  To use explicit values, the array must have exactly two entries. |

flip\_horiz

|  |  |
| --- | --- |
| Type | bool |
| Description | If true, the images being polled will be horizontally mirrored. |
| Values | true to mirror the polled images horizontally. Else, their horizontalness will be left alone. |
| Notes | Depending on the internal or low-level support of the camera or its drivers, this may (or may not) incur compute overhead. |

flip\_vert

|  |  |
| --- | --- |
| Type | Bool |
| Description | If true, the images being polled will be vertically mirrored. |
| Values | true to mirror the polled images vertically. Else, their verticalness will be left alone. |
| Notes | Depending on the internal or low-level support of the camera or its drivers, this may (or may not) incur compute overhead. |

menu\_targ

|  |  |
| --- | --- |
| Type | bool |
| Description | Used to specify if the menu is the modification target of the applications camera UI. |
| Values | true to show the application fullscreen (default).  false to show the application windowed. |
| Notes | Only 1 camera can be the menu target. If multiple things are set to true, the first found entry in *feed\_options* will be used. |

pipe\_chans

|  |  |
| --- | --- |
| Type | int |
| Description | Used when the polling method(e.g., *default\_poll*) is set to “external”. |
| Values | Used to specify the number if color channels per pixel |
| Notes | Has not been recently tested.  Piped streaming doesn’t have a handshaking process that shares the video stream information, so that must be explicitly defined. |

pipe\_cmd

|  |  |
| --- | --- |
| Type | string |
| Description | Used when polling method (e.g., *default\_poll*) is set to “external”. |
| Values | The command to execute for a video streaming implementation using pipes.  The application referenced should spit out raw binary luminosity or RGB data with its standard output stream. |
| Notes | Has not been recently tested. |

pipe\_height

|  |  |
| --- | --- |
| Type | int |
| Description | The image height to use when the polling method is set to “external”. |
| Values | The height of the images being streamed from the external program’s output pipe. |
| Notes | Has not been recently tested.  Piped streaming doesn’t have a handshaking process that shares the video stream information, so that must be explicitly defined. |

pipe\_width

|  |  |
| --- | --- |
| Type | int |
| Description | The image width to use when the polling method is set to “external”. |
| Values | The width of the images being streamed from the external program’s output pipe. |
| Notes | Has not been recently tested.  Piped streaming doesn’t have a handshaking process that shares the video stream information, so that must be explicitly defined. |

processing

|  |  |
| --- | --- |
| Type | string, or int |
| Description | The type of image process to apply to the video stream images. |
| Values | If an int is used, it is implied that a *static\_threshold* should be used, with the integer (a value between [0, 255]) used as the threshold pixel value. If a string is specified, it should be one of the string values listed below.  Used to specify the type of image processing to use on the video frames:   * **“none”**   + Do not apply image processing. * **“yen\_threshold”**   + Use a threshold calculated from Yen’s algorithm. This is currently using a process similar to the original CVG GAIN application. * **“yen\_threshold\_compressed”**   + Use a threshold calculated from Yen’s algorithm. This is a simplified version that optimizes out expensive and low-impact parts of the yen\_threshold process. * **“static\_threshold”**   + For development – use a PNG image as a placeholder camera string. This gets rid of the dependency for a camera, creates a static unchanging benchmark image, and does not require camera initialization time. * **“two\_stdev\_from\_mean”**   + A threshold method calculates the mean and thresholds using a value that’s two standard deviations below that mean. |
| Notes | -- |

static\_img

*This documented value is a deprecated feature that is considered for deletion. It is currently under review if we are going to commit with its obsolescence or repurpose the feature in question. Until then, because it’s saved in the json file, it’s documented for the sake of completeness.*

|  |  |
| --- | --- |
| Type | string |
| Description | Unused |
| Values | Unused |
| Notes | Unused |

stream\_width

|  |  |
| --- | --- |
| Type | int |
| Description | The explicitly defined size, in pixels, of what the width of the streaming video feed should be. |
| Values | Pixel amount. |
| Notes | Depending on the polling implementation, this may be treated more as a preference than a hard rule.  Set to 0 to use the implementation default. |

stream\_height

|  |  |
| --- | --- |
| Type | Int |
| Description | The explicitly defined size, in pixels, of what the height of the streaming video feed should be. |
| Values | Pixel amount. |
| Notes | Depending on the polling implementation, this may be treated more as a preference than a hard rule.  Set to 0 to use the implementation default. |

uri

|  |  |
| --- | --- |
| Type | string |
| Description | Used when *default\_poll* is set to “url”.  This will be the web resource to stream video from. At the moment, only RTSP (via v4l2-rtspserver) has been tested to work. |
| Values | URL to internet streaming video supported by OpenCV (with V4L2 and MMPEG implementations enabled). |
| Notes | -- |

video\_exposure\_micro

|  |  |
| --- | --- |
| Type | Int |
| Description | The exposure time for capturing camera frames. |
| Values | The amount of time, in microseconds. |
| Notes | Set to 0 to use the implementation default. This will probably be around 30FPS but will depend on the specific implementation used.  How supported and accurate this value is, will depend on the implementation. |

## Terms

### Screen Coordinate System

A screen coordinate system, defined in pixels, where the origin (unless specified otherwise) will be the top left of the screen. The positive X axis will point right. The positive Y axis will point down.