Lambda Vue

2.0.10

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 lambdavue Namespace Reference

This is the header file where API functions of LambdaVue SDK is contained. This file contains API functions decalaration.

Classes

- struct EventValue

  A struct about LambdaVue event This is the structure of callback event value. Callback event have different type. You have to check the value type before any access or you may get wrong value (e.g. the callback value is string but you access integer).
- struct FilterParameter

  A struct about parameters of algorithm, store all the parameter value.
- class Magnifier

  This class provides some API functions to control the LambdaVue magnifier.
- class MediaWriter

  This class provides some API functions control LambdaVue media writer.
- class SourceManager

  This class provides some API functions to management LambdaVue video source.
- struct VerificationResult

  A struct to describe license verification result.
- class WriterManager

  This class provides some API functions to management LambdaVue media writer WriterManager contains two MediaWriter, one for original video and the other for processed video.
- struct WriterState

  A struct to describe current video writer status.

Typedefs

- typedef void(∗ CallBackFunction) (struct EventValue)

  Callback function.

Enumerations

- enum EventType {
  NONE_EVENT = -1, ORIGINAL_IMAGE = 0, PROCESSED_IMAGE = 1, ROI_RESET = 2,
  FILTER_CHANGED = 3, COMPOSE_MODE_CHANGED = 4, VALUE_ALPHA = 5, VALUE_LOWCUT = 6,
  VALUE_HIGHCUT = 7, VALUE_LAMBDA = 8, VALUE_LEVEL = 9, VALUE_SOURCE_FRAME_RATE = 10,
  VALUE_PROCESS_FRAME_RATE = 11, VALUE_FILE_DURATION = 12, VALUE_FILE_PROGRESS = 13,
  VALUE_CAMERA_COUNT = 14,
  AUTOFOCUS_AVAILABLE = 15, AUTOFOCUS_ENABLED = 16, OPEN_CAMERA_SUCCESS = 17, OP↩
  EN_CAMERA_FAIL = 18,
  OPEN_FILE_SUCCESS = 19, OPEN_FILE_FAIL = 20, CLOSE_SOURCE_SUCCESS = 21, FILE_ARCHI↩
  VED = 22,
  END_OF_FILE = 23, DEBUG_MESSAGE = 24, INFO_MESSAGE = 25, WARNING_MESSAGE = 26,
  ERROR_MESSAGE = 27 }

  An enumerations about LambdaVue Callback Events.
- enum ValueType {
  NON_VALUE, INTEGER_VALUE, FLOAT_VALUE, STRING_VALUE,
  FRAME_VALUE }

  The callback event value type.
- enum ProcessMethod { **NO_METHOD** = -1, LINEAR_MOTION = 0, LINEAR_COLOR = 1, RIESZ_MOTION = 2 }

  An enumerations about filter algorithm.
- enum SourceOperation { **CLOSE_SOURCE** = 0, OPEN_CAMERA_SOURCE = 1, OPEN_FILE_SOURCE = 2 }

  An enumerations about video source opening and closing.
- enum SourceAccess {
  CAMERA_EXPOSURE = 0, CAMERA_AUTO_FOCUS = 1, CAMERA_FRAME_RATE = 2, CAMERA_BR↩
  IGHTNESS = 3,
  CAMERA_CONTRAST = 4, CAMERA_SATURATION = 5, CAMERA_AUTO_WHITEBALANCE = 6, CAM↩
  ERA_BACKLIGHT_COMPENSATION = 7,
  CAMERA_ABSOLUTE_FOCUS = 8, CAMERA_ZOOM_IN = 9, CAMERA_ZOOM_OUT = 10, CAMERA_Z↩
  OOM_RESET = 11,
  CAMERA_PAN_LEFT = 12, CAMERA_PAN_RIGHT = 13, CAMERA_PAN_RESET = 14, CAMERA_TILT↩
  _UP = 15,
  CAMERA_TILT_DOWN = 16, CAMERA_TILT_RESET = 17, CAMERA_PTZ_RESET = 18, FILE_SEEK = 100,
  FILE_PROGRESS = 101, FILE_DURATION = 102, FILE_PLAY = 103, FILE_PAUSE = 104,
  FILE_FORWARD = 105, FILE_BACKWARD = 106, FILE_STOP = 107, FILE_NORMAL_PLAYBACK = 108,
  FILE_TURBO_PLAYBACK = 109, FILE_LOOP_PLAY = 110, FILE_SINGLE_PLAY = 111, FILE_RECORD↩
  _SEEK = 112 }

  An enumerations about video source contorl/query type, only some of them are available.
- enum ParameterType {
  ALPHA = 0, HIGHCUT = 1, LOWCUT = 2, LEVEL = 3,
  LEVEL_BOUND = 4, LAMBDA = 5, LAMBDA_BOUND = 6, FRAMERATE = 7,
  CUTOFF_BOUND = 8 }

  An enumerations about parameters type of filter algorithm.
- enum RecordType { **RECORD_NONE** = 0, RECORD_ORIGINAL_ONLY = 1, RECORD_PROCESSED_O↩
  NLY = 2, RECORD_BOTH = 3 }

  An enumerations about video record types.
- enum ComposeMode { **COMPOSE_NONE** = 0, COMPOSE_PAP = 1, COMPOSE_PIP_MAIN_PROCESS↩
  ED = 2, COMPOSE_PIP_MAIN_ORIGINAL = 3 }

  An enumerations about video compose modes.
- enum OutputFileType { **OUTPUT_BOTH** = 0, OUTPUT_ORIGINAL = 1, OUTPUT_PROCESSED = 2 }

  An enumerations about output video file types.
- enum LicenseStatus { **LICENSE_OFFLINE_WARNING** = -2, LICENSE_OFFLINE_SUCCESS = -1, LICE↩
  NSE_SUCCESS = 0, LICENSE_FAILURE = 1 }

An enumerations about LambdaVue license verification status.

- enum LicenseErrorCode {
  LICENSE_ERROR_NONE = 0, **LICENSE_ERROR_MISSING_LICENSE** = 201, **LICENSE_ERROR_INC↩ OMPLETE_LICENSE** = 202, **LICENSE_ERROR_INVALID_SIGNATURE_LOCAL** = 203, **LICENSE_ERROR_INVALID_SIGNATURE_REMOTE** = 204, **LICENSE_ERROR_MISMATCHED_MAC** = 205, **LICENSE_ERROR_EXPIRED** = 206, **LICENSE_ERROR_TIMESTAMP_MODIFIED** = 207, **LICENSE_ERROR_INVALID_ACTIVATION_CODE** = 208, **LICENSE_ERROR_OFFLINE_FORBIDDEN** = 209, **LICENSE_ERROR_MISMATCHED_PRODUCT_TYPE** = 210 }

  An enumerations to describe LambdaVue license verification error.

Functions

- struct VerificationResult InitMagEngine (CallBackFunction cb, const char *product="SDK2")

  Initialize LambdaVue magnification engine.
- time_t GetExpirationDate ()

  Get the expired date of your LambdaVue license.
- std::string GetVerificationErrorString (enum LicenseErrorCode verification_error_code)

  Return string containing an verification error string corresponding to the verification error code.
- std::string GetLast5CharOfActivationCode ()

  Return string containing last 5 character of users activation code.
- std::string GetActivationCode ()

  Return string containing users activation code.
- void DestroyEngine ()

  Release LambdaVue magnification engine.
- void OpenSource (enum SourceOperation operation, const char *source, bool idle=true)

  Open video file or a capturing device as source video Set the video source form a file or webcam if the device is a webcam, use videoX, X is the camera index.
- void CloseSource (bool idle=true)

  Close the video source.
- void SetBypassMagnify (bool bypass)

  Setting whether to bypass magnify stage.
- void SetOutputFilename (enum OutputFileType type, const char *filename)

  Setting the filename of output videos.
- void SetRecordType (enum RecordType type)

  Reccord type setting.
- struct WriterState GetWriterState ()

  Get the current state of video writer.
- int GetCameraCount ()

  Get the number of connected cameras.
- bool ControlSource (enum SourceAccess type, int value=0)

  Contorl video source.
- int64_t QuerySource (enum SourceAccess type)

  Query the ability or status of video source.
- std::string GetSourceName ()

  Get the name of video source.
- void SetProcessMethod (enum ProcessMethod method)

  Filter algorithm setting function Set video filtering method to specified method (linear motion, linear color, Riesz motion)
- enum ProcessMethod GetProcessMethod ()

  Get Current filter algorithm.
- void SetComposeMode (enum ComposeMode mode)

Compose mode setting function Set video compose mode to specified mode (none, pap, pip)

- enum ComposeMode GetComposeMode ()

  Get Current compose mode.

- void SetFilterParameter (enum ParameterType type, float value)

  Filter parameter setting.

- struct FilterParameter GetFilterParameter ()

  Get the FilterParamter structure The FilterParameter structure contains the current value of parameters.

- void SetROIView (uint16_t x, uint16_t y, uint16_t width, uint16_t height)

  Set the region of interest.

- std::string LAMBDAVUE_SDK_VERSION ()

  Get the version of LambdaVue SDK.

3.1.1 Detailed Description

This is the header file where API functions of LambdaVue SDK is contained. This file contains API functions decalaration.

LmabdaVue.h

Author

Nathan Chen

Copyright

3.1.2 Enumeration Type Documentation

3.1.2.1 enum lambdavue::ComposeMode

An enumerations about video compose modes.

Enumerator

COMPOSE_PAP Do not compose video

COMPOSE_PIP_MAIN_PROCESSED Picture-and-Picture, original video at left side

COMPOSE_PIP_MAIN_ORIGINAL Picture-in-Picture, use processed video as main picutre

3.1.2.2 enum lambdavue::EventType

An enumerations about LambdaVue Callback Events.

@/definedblock

Enumerator

ORIGINAL_IMAGE None event

PROCESSED_IMAGE Evoked when source image is captured

ROI_RESET Evoked when processed image is done

FILTER_CHANGED Evoked when region of interest had been changed

COMPOSE_MODE_CHANGED Evoked when filter method had been changed

VALUE_ALPHA Evoked when compose mode had been changed

VALUE_LOWCUT Evoked when magnification value had been changed

VALUE_HIGHCUT Evoked when filter low cutoff had been changed

VALUE_LAMBDA Evoked when filter high cutoff had been changed

VALUE_LEVEL Evoked when spatial frequency cutoff had been changed (used in linear motion only)

VALUE_SOURCE_FRAME_RATE Evoked when Gaussian pyramid level had been changed (used in linear color only)

VALUE_PROCESS_FRAME_RATE Evoked when source frame rate had been changed

VALUE_FILE_DURATION Evoked when processing frame rate had been changed

VALUE_FILE_PROGRESS Evoked when get video file length

VALUE_CAMERA_COUNT Evoked during file processing, return the timestamp in video file

AUTOFOCUS_AVAILABLE Evoked when camera count had been changed

AUTOFOCUS_ENABLED Evoked after source opened, indicate whether the source can do auto focus

OPEN_CAMERA_SUCCESS Evoked after source opened, indicate whether auto focus is enabled

OPEN_CAMERA_FAIL Evoked when webcam opened successful

OPEN_FILE_SUCCESS Evoked when webcam opened fail

OPEN_FILE_FAIL Evoked when file opened successful

CLOSE_SOURCE_SUCCESS Evoked when file opened fail

FILE_ARCHIVED Evoked when source closed successful

END_OF_FILE Evoked when file archived

DEBUG_MESSAGE Evoked when reach to end of file

INFO_MESSAGE Evoked when debug event occurred

WARNING_MESSAGE Evoked when info event occurred

ERROR_MESSAGE Evoked when warning event occurred

3.1.2.3 enum lambdavue::LicenseStatus

An enumerations about LambdaVue license verification status.

Enumerator

LICENSE_OFFLINE_SUCCESS The LambdaVue license has passed offline verification, but must take an online verification in 7 days

LICENSE_SUCCESS The LambdaVue license has passed offline verification

LICENSE_FAILURE The LambdaVue license has passed online verification

3.1.2.4 enum lambdavue::OutputFileType

An enumerations about output video file types.

Enumerator

OUTPUT_ORIGINAL Both origianl and processed video

OUTPUT_PROCESSED The origianl video

3.1.2.5 enum lambdavue::ParameterType

An enumerations about parameters type of filter algorithm.

Enumerator

HIGHCUT The amplification factor

LOWCUT High cutoff of temporal filter

LEVEL Low cutoff of temporal filter

LEVEL_BOUND Gaussian pyramid level (used in linear color only)

LAMBDA Maximum value for level

LAMBDA_BOUND Spatial frequency cutoff (used in linear motion only)

FRAMERATE Maximum value for lambda

CUTOFF_BOUND Framerate of video source

3.1.2.6 enum lambdavue::ProcessMethod

An enumerations about filter algorithm.

Enumerator

LINEAR_MOTION No method selected

LINEAR_COLOR Eulerian linear motion amplification method

RIESZ_MOTION Eulerian linear color amplification method

3.1.2.7 enum lambdavue::RecordType

An enumerations about video record types.

Enumerator

RECORD_ORIGINAL_ONLY Do not record any video

RECORD_PROCESSED_ONLY Only record original video

RECORD_BOTH Only record processed video

3.1.2.8 enum lambdavue::SourceAccess

An enumerations about video source contorl/query type, only some of them are available.

Enumerator

CAMERA_AUTO_FOCUS Camera exposure access
CAMERA_FRAME_RATE Camera auto focus access
CAMERA_BRIGHTNESS Camera frame rate access
CAMERA_CONTRAST Camera brightness access
CAMERA_SATURATION Camera contrast access
CAMERA_AUTO_WHITEBALANCE Camera saturation access
CAMERA_BACKLIGHT_COMPENSATION Camera whitebalance access
CAMERA_ABSOLUTE_FOCUS Camera backlight compensation access
CAMERA_ZOOM_IN Camera absolute focus access
CAMERA_ZOOM_OUT Camera zoom in
CAMERA_ZOOM_RESET Camera zoom out
CAMERA_PAN_LEFT Camera zoom reset
CAMERA_PAN_RIGHT Camera pan left
CAMERA_PAN_RESET Camera pan right
CAMERA_TILT_UP Camera pan reset
CAMERA_TILT_DOWN Camera tilt up
CAMERA_TILT_RESET Camera tilt down
CAMERA_PTZ_RESET Camera tilt reset
FILE_SEEK Reset pan, tilt and zoom
FILE_PROGRESS Video file seek to some timestamp
FILE_DURATION Video file current progress
FILE_PLAY Video file length
FILE_PAUSE Play video file
FILE_FORWARD Pause video file
FILE_BACKWARD Foward video file
FILE_STOP Backward video file
FILE_NORMAL_PLAYBACK Stop video file
FILE_TURBO_PLAYBACK Video file normal playback
FILE_LOOP_PLAY Video file turbo playback
FILE_SINGLE_PLAY Loop play video file
FILE_RECORD_SEEK Play video file once

3.1.2.9 enum lambdavue::SourceOperation

An enumerations about video source opening and closing.

Enumerator

OPEN_CAMERA_SOURCE Close video source
OPEN_FILE_SOURCE Open webcam source

3.1.2.10 enum lambdavue::ValueType

The callback event value type.

Enumerator

INTEGER_VALUE NON_VALUE

FLOAT_VALUE INTEGER_VALUE

STRING_VALUE FLOAT_VALUE

FRAME_VALUE STRING_VALUE
FRAME_VALUE

3.1.3 Function Documentation

3.1.3.1 bool lambdavue::ControlSource (enum SourceAccess *type,* int *value* = 0)

Contorl video source.

Parameters

type	Specify control type
value	Specify the value

Returns

bool Whether the control is successful

3.1.3.2 std::string lambdavue::GetActivationCode ( )

Return string containing users activation code.

Returns

std::string The users activation code

3.1.3.3 int lambdavue::GetCameraCount ( )

Get the number of connected cameras.

Returns

int The number of connected cameras

3.1.3.4 enum ComposeMode lambdavue::GetComposeMode ( )

Get Current compose mode.

Returns

enum ComposeType The current chosen mode

3.1.3.5 time_t lambdavue::GetExpirationDate ( )

Get the expired date of your LambdaVue license.

Returns

time_t The Expired date

3.1.3.6 struct FilterParameter lambdavue::GetFilterParameter ( )

Get the FilterParamter structure The FilterParameter structure contains the current value of parameters.

Returns

struct FilterParameter The current filter parameter struct

3.1.3.7 std::string lambdavue::GetLast5CharOfActivationCode ( )

Return string containing last 5 character of users activation code.

Returns

std::string The last 5 character of users activation code

3.1.3.8 enum ProcessMethod lambdavue::GetProcessMethod ( )

Get Current filter algorithm.

Returns

enum ProcessMethod The current chosen algorithm

3.1.3.9 std::string lambdavue::GetSourceName ( )

Get the name of video source.

Returns

std::string Current source name

3.1.3.10 std::string lambdavue::GetVerificationErrorString (enum LicenseErrorCode *verification_error_code*)

Return string containing an verification error string corresponding to the verification error code.

Parameters

verification_error_code	Error code to describe

Returns

std::string The verification error string

3.1.3.11 struct WriterState lambdavue::GetWriterState ( )

Get the current state of video writer.

Returns

struct WriterState Current video writer status

3.1.3.12 struct VerificationResult lambdavue::InitMagEngine (CallBackFunction *cb*, const char * *product* = "SDK2")

Initialize LambdaVue magnification engine.

Parameters

cb	Pass your CallBackFuntion as a parameter

Returns

enum LicenseStatus The status of your LambdaVue license

3.1.3.13 std::string lambdavue::LAMBDAVUE_SDK_VERSION ( )

Get the version of LambdaVue SDK.

Returns

std::string The version of LambdaVue SDK

3.1.3.14 void lambdavue::OpenSource (enum SourceOperation *operation*, const char * *source*, bool *idle* = true)

Open video file or a capturing device as source video Set the video source form a file or webcam if the device is a webcam, use videoX, X is the camera index.

Parameters

operation	Specify the SourceOperation
source	The name of the video source (filename or device name)
idle	Default value is true, set this parameter to false only if you are developing an OSX console program

3.1.3.15 int64_t lambdavue::QuerySource (enum SourceAccess *type*)

Query the ability or status of video source.

Parameters

type	Specify control type

Returns

int64_t The current value of the query type, if the type is unaccessible then UNAVAILABLE is returned

3.1.3.16 void lambdavue::SetBypassMagnify (bool *bypass*)

Setting whether to bypass magnify stage.

Parameters

bypass	Specify whether to bypass magnify stage

3.1.3.17 void lambdavue::SetComposeMode (enum ComposeMode *mode*)

Compose mode setting function Set video compose mode to specified mode (none, pap, pip)

Parameters

mode	The compose mode

3.1.3.18 void lambdavue::SetFilterParameter (enum ParameterType *type,* float *value*)

Filter parameter setting.

Parameters

type	Specify parameter type
value	Parameter value

3.1.3.19 void lambdavue::SetOutputFilename (enum OutputFileType *type,* const char * *filename*)

Setting the filename of output videos.

Parameters

type	Specify the output file (original or processed) type you'd like to named it
filename	Specify the file name

3.1.3.20 void lambdavue::SetProcessMethod (enum ProcessMethod *method*)

Filter algorithm setting function Set video filtering method to specified method (linear motion, linear color, Riesz motion)

Parameters

method	The filter algorithm

3.1.3.21 void lambdavue::SetRecordType (enum RecordType *type*)

Reccord type setting.

Parameters

type	Specify the record type

3.1.3.22 void lambdavue::SetROIView (uint16_t *x,* uint16_t *y,* uint16_t *width,* uint16_t *height*)

Set the region of interest.

Parameters

x	The x coordinate of the top-left corner of ROI
y	The y coordinate of the top-left corner of ROI
width	The width of ROI
height	The height of ROI

Chapter 4

Class Documentation

4.1 lambdavue::EventValue Struct Reference

A struct about LambdaVue event This is the structure of callback event value. Callback event have different type. You have to check the value type before any access or you may get wrong value (e.g. the callback value is string but you access integer).

`#include <core.h>`

Public Member Functions

- EventValue (EventType evt=NONE_EVENT)
- EventValue (EventType evt, bool i)
- EventValue (EventType evt, int i)
- EventValue (EventType evt, size_t i)
- EventValue (EventType evt, int64_t i)
- EventValue (EventType evt, float d)
- EventValue (EventType evt, double d)
- EventValue (EventType evt, std::string s)
- EventValue (EventType evt, char *data, uint32_t len, uint16_t w, uint16_t h)

Public Attributes

- EventType **event**
- ValueType **value_type**
- std::string **s_value**
- union {
  double **d_value**
  int64_t **i_value**
  };

- void * **data_buffer**
- uint16_t **image_width**
- uint16_t **image_height**

4.1.1 Detailed Description

A struct about LambdaVue event This is the structure of callback event value. Callback event have different type. You have to check the value type before any access or you may get wrong value (e.g. the callback value is string but you access integer).

event A enum EventType, indicate the type of event value_type The callback event value type s_value String value type d_value Double value type i_value Integer int64_t value type data_buffer Pointer to data buffer which contains image data image_width Image width image_height Image height

4.1.2 Constructor & Destructor Documentation

4.1.2.1 lambdavue::EventValue::EventValue (EventType *evt* = NONE_EVENT) [inline]

Constructor for no value event type

4.1.2.2 lambdavue::EventValue::EventValue (EventType *evt*, bool *i*) [inline]

Constructor for boolean type

4.1.2.3 lambdavue::EventValue::EventValue (EventType *evt*, int *i*) [inline]

Constructor for integer type

4.1.2.4 lambdavue::EventValue::EventValue (EventType *evt*, size_t *i*) [inline]

Constructor for unsigned integer type

4.1.2.5 lambdavue::EventValue::EventValue (EventType *evt*, int64_t *i*) [inline]

Constructor for 64bits integer typev

4.1.2.6 lambdavue::EventValue::EventValue (EventType *evt*, float *d*) [inline]

Constructor for floating point type

4.1.2.7 lambdavue::EventValue::EventValue (EventType *evt*, double *d*) [inline]

Constructor for double floating point type

4.1.2.8 lambdavue::EventValue::EventValue (EventType *evt*, std::string *s*) [inline]

Constructor for string type

4.1.2.9 lambdavue::EventValue::EventValue (EventType *evt*, char * *data*, uint32_t *len*, uint16_t *w*, uint16_t *h*) [inline]

Constructor for image buffer type

The documentation for this struct was generated from the following file:

- LambdaVue/core/include/core.h

4.2 lambdavue::FilterParameter Struct Reference

A struct about parameters of algorithm, store all the parameter value.

`#include <core.h>`

Public Attributes

- size_t **alpha**
- float **high_cutoff**
- float **low_cutoff**
- float **lambda**
- size_t **level**
- float **framerate**
- float **lambda_bound**
- float **cutoff_bound**

4.2.1 Detailed Description

A struct about parameters of algorithm, store all the parameter value.

alpha The amplification factor high_cutoff High cutoff of temporal filter low_cutoff Low cutoff of temporal filter lambda Spatial frequency cutoff level Gaussian pyramid level framerate Framerate of video source lambda_bound Maximum value for lambda cutoff_bound Maximum value for temporal filter cutoff

The documentation for this struct was generated from the following file:

- LambdaVue/core/include/core.h

4.3 lambdavue::Magnifier Class Reference

This class provides some API functions to control the LambdaVue magnifier.

`#include <magnifier.h>`

Public Member Functions

- Magnifier (const char *product)

  Constructor of magnifier.
- ~Magnifier ()

  Destructor of magnifier.
- void MagDeleteFilter ()

  Delete the filter of LambdaVue magnifier.
- bool MagHaveFilter ()

  Return whether the LambdaVue magnifier has any filter.
- void MagSetProcessMethod (enum ProcessMethod method, float framerate=-1)

  LambdaVue magnifer filter algorithm setting function.
- enum ProcessMethod MagGetProcessMethod ()

  Get Current filter algorithm of LambdaVue magnifier.
- void MagSetComposeMode (enum ComposeMode mode)

  LambdaVue magnifer compose mode setting function.
- enum ComposeMode MagGetComposeMode ()

  Get Current compose mode of LambdaVue magnifier.
- void MagSetFilterParameter (enum ParameterType type, float value)

  LambdaVue magnifer filter parameter setting.
- struct FilterParameter MagGetFilterParameter ()

  Get the FilterParamter structure of LambdaVue magnifier The FilterParameter structure contains the current value of parameters.
- void MagPushImage (BufferStruct &source)

  Push an image into LambdaVue magnifier.
- void MagGetImage (BufferStruct &result)

  Return the processed image from LambdaVue magnifier.
- void MagSetRoi (uint16_t x, uint16_t y, uint16_t width, uint16_t height)

  Set the region of interest of LambdaVue magnifier.
- struct VerificationResult MagGetLicenseVerificationResult ()

  Return license verification result.
- time_t MagGetExpiryTimeT ()

  Get the expired date of your LambdaVue license.
- std::string MagGetLast5CharActivationCode ()

  Get last 5 character of your LambdaVue activation code.
- std::string MagGetActivationCode ()

  Get your LambdaVue activation code.

4.3.1 Detailed Description

This class provides some API functions to control the LambdaVue magnifier.

4.3.2 Member Function Documentation

4.3.2.1 std::string lambdavue::Magnifier::MagGetActivationCode ( )

Get your LambdaVue activation code.

Returns

std::string The activation code

4.3.2.2 enum ComposeMode lambdavue::Magnifier::MagGetComposeMode ( )

Get Current compose mode of LambdaVue magnifier.

Returns

enum ComposeMode The current compose mode of LambdaVue magnifier

4.3.2.3 time_t lambdavue::Magnifier::MagGetExpiryTimeT ( )

Get the expired date of your LambdaVue license.

Returns

time_t The Expired date

4.3.2.4 struct FilterParameter lambdavue::Magnifier::MagGetFilterParameter ( )

Get the FilterParamter structure of LambdaVue magnifier The FilterParameter structure contains the current value of parameters.

Returns

struct FilterParameter The current filter parameter struct

4.3.2.5 void lambdavue::Magnifier::MagGetImage (BufferStruct & *result*)

Return the processed image from LambdaVue magnifier.

Returns

BufferStruct The BufferStruct of processed image

4.3.2.6 std::string lambdavue::Magnifier::MagGetLast5CharActivationCode ( )

Get last 5 character of your LambdaVue activation code.

Returns

std::string The last 5 character of activation code

4.3.2.7 struct VerificationResult lambdavue::Magnifier::MagGetLicenseVerificationResult ( )

Return license verification result.

Returns

struct VerificationResult The verification result struct of your LambdaVue license

4.3.2.8 enum ProcessMethod lambdavue::Magnifier::MagGetProcessMethod ( )

Get Current filter algorithm of LambdaVue magnifier.

Returns

enum ProcessMethod The current chosen algorithm of LambdaVue magnifier

4.3.2.9 bool lambdavue::Magnifier::MagHaveFilter ( )

Return whether the LambdaVue magnifier has any filter.

Returns

bool Whether the magnifier has any filter

4.3.2.10 void lambdavue::Magnifier::MagPushImage (BufferStruct & *source*)

Push an image into LambdaVue magnifier.

Parameters

source	The image BufferStruct

4.3.2.11 void lambdavue::Magnifier::MagSetComposeMode (enum ComposeMode *mode*)

LambdaVue magnifer compose mode setting function.

Parameters

mode	The compose mode

4.3.2.12 void lambdavue::Magnifier::MagSetFilterParameter (enum ParameterType *type*, float *value*)

LambdaVue magnifer filter parameter setting.

Parameters

type	Specify parameter type
value	Parameter value

4.3.2.13 void lambdavue::Magnifier::MagSetProcessMethod (enum ProcessMethod *method*, float *framerate* = −1)

LambdaVue magnifer filter algorithm setting function.

Parameters

method	The filter algorithm
framerate	The framerate of video source

4.3.2.14 void lambdavue::Magnifier::MagSetRoi (uint16_t *x*, uint16_t *y*, uint16_t *width*, uint16_t *height*)

Set the region of interest of LambdaVue magnifier.

Parameters

x	The x coordinate of the top-left corner of ROI
y	The y coordinate of the top-left corner of ROI
width	The width of ROI
height	The height of ROI

The documentation for this class was generated from the following file:

- /Users/nathanchen/Desktop/LambdaVue/LambdaVueSDK/LambdaVue/magnifier/include/magnifier.h

4.4 lambdavue::MediaWriter Class Reference

This class provides some API functions control LambdaVue media writer.

`#include <writer.hpp>`

Public Member Functions

- MediaWriter ()

  Constructor of media writer.
- ~MediaWriter ()

  Destructor of media writer.
- void MwOpenNewFile ()

  Prepare a new output video file.
- void MwClose ()

Close the output video and delete the encoder.

- void MwSetAV (int width, int height, float framerate, int sample_rate, int channel_number)

  Set the basic information of audio (not yet supported) and video for media writer.
- void MwSetFileName (std::string path, std::string filename)

  Set the filename of output video.
- void MwWriteVideo (lambdavue::buffer image, int64_t timestamp)

  Encode an image buffer and write to video file.
- void MwSetWidth (int width)

  Set the width of the output video.
- int MwGetWidth ()

  Return the width of the output video.
- void MwSetHeight (int height)

  Set the height of the output video.
- int MwGetHeight ()

  Return the height of the output video.
- int MwGetSampleRate ()

  Return the audio sample rate of the output video.
- int MwGetChannelNumber ()

  Return the number of channels of the output video.
- std::string MwGetDestPath ()

  Return the destination path of the output video.
- std::string MwGetDestFilename ()

  Return the filename of the output video.
- std::string MwGetDestFullFilename ()

  Return the full filename (path and filename) of the output video.

4.4.1 Detailed Description

This class provides some API functions control LambdaVue media writer.

4.4.2 Member Function Documentation

4.4.2.1 int lambdavue::MediaWriter::MwGetChannelNumber ( ) [inline]

Return the number of channels of the output video.

Returns

int The number of channels of the output video

4.4.2.2 std::string lambdavue::MediaWriter::MwGetDestFilename ( ) [inline]

Return the filename of the output video.

Returns

std::string The filename of output video

4.4.2.3 std::string lambdavue::MediaWriter::MwGetDestFullFilename ( ) [inline]

Return the full filename (path and filename) of the output video.

Returns

std::string The full filename (path and filename) of output video

4.4.2.4 std::string lambdavue::MediaWriter::MwGetDestPath ( ) [inline]

Return the destination path of the output video.

Returns

std::string The destination path of output video

4.4.2.5 int lambdavue::MediaWriter::MwGetHeight ( ) [inline]

Return the height of the output video.

Returns

int The height of the output video

4.4.2.6 int lambdavue::MediaWriter::MwGetSampleRate ( ) [inline]

Return the audio sample rate of the output video.

Returns

int The audio sample rate of the output video

4.4.2.7 int lambdavue::MediaWriter::MwGetWidth ( ) [inline]

Return the width of the output video.

Returns

int The width of the output video

4.4.2.8 void lambdavue::MediaWriter::MwSetAV (int *width,* int *height,* float *framerate,* int *sample_rate,* int *channel_number*)

Set the basic information of audio (not yet supported) and video for media writer.

Parameters

width	Width of output video
height	Height of output video
framerate	Framerate of output video
sample_rate	Audio sample rate of output video (not yet supported)
channel_number	Channel number of output video (not yet supported)

4.4.2.9 void lambdavue::MediaWriter::MwSetFileName (std::string *path,* std::string *filename*)

Set the filename of output video.

Parameters

path	Destination path of output video
filename	Filename of output video

4.4.2.10 void lambdavue::MediaWriter::MwSetHeight (int *height*)

Set the height of the output video.

Returns

int The height of the output video

4.4.2.11 void lambdavue::MediaWriter::MwSetWidth (int *width*)

Set the width of the output video.

Returns

int The width of the output video

4.4.2.12 void lambdavue::MediaWriter::MwWriteVideo (lambdavue::buffer *image,* int64_t *timestamp*)

Encode an image buffer and write to video file.

Parameters

img	Image buffer
timestamp	The timestamp of the image buffer

The documentation for this class was generated from the following file:

- /Users/nathanchen/Desktop/LambdaVue/LambdaVueSDK/LambdaVue/writer/include/writer.hpp

4.5 lambdavue::SourceManager Class Reference

This class provides some API functions to management LambdaVue video source.

`#include <source.h>`

Public Member Functions

- SourceManager ()

  Constructor of source manager.
- ∼SourceManager ()

  Destructor of source manager.
- bool SmSetSource (enum SourceOperation operation, const char ∗source)

  Video source setting or closing.
- void SmGetImage (BufferStruct &output_frame)

  Get the decoded frame image from LambdaVue source manager.
- std::string SmGetSourceName ()

  Get the name of video source.
- size_t SmGetFrameCount ()

  Return the amount of frame in video source.
- size_t SmGetWidth ()

  Return the image width of video source.
- size_t SmGetHeight ()

  Return the image height of video source.
- float SmGetFramerate ()

  Return the framerate of video source.
- int SmGetCameraCount ()

  Return the amount of connected camera.
- bool sm_is_valid ()

  Return whether the source manager is valid.
- bool sm_is_file ()

  Return whether the current video source is file.
- int64_t SmQuery (enum lambdavue::SourceAccess type)

  Query the ability or status of video source.
- bool SmControl (enum lambdavue::SourceAccess type, int value)

  Contorl video source.

4.5.1 Detailed Description

This class provides some API functions to management LambdaVue video source.

4.5.2 Member Function Documentation

4.5.2.1 bool lambdavue::SourceManager::sm_is_file ( )

Return whether the current video source is file.

Returns

bool True if the source is a file, false otherwise

4.5.2.2 bool lambdavue::SourceManager::sm_is_valid ( )

Return whether the source manager is valid.

Returns

bool Whether the source manager is valid

4.5.2.3 bool lambdavue::SourceManager::SmControl (enum **lambdavue::SourceAccess** *type,* int *value*)

Contorl video source.

Parameters

type	Specify control type
value	Specify the value

Returns

bool Whether the control is successful

4.5.2.4 int lambdavue::SourceManager::SmGetCameraCount ( )

Return the amount of connected camera.

Returns

int The total number of connected camera

4.5.2.5 size_t lambdavue::SourceManager::SmGetFrameCount ( )

Return the amount of frame in video source.

Returns

size_t The total number of frame in video source

4.5.2.6 float lambdavue::SourceManager::SmGetFramerate ( )

Return the framerate of video source.

Returns

float The framerate of video source

4.5.2.7 size_t lambdavue::SourceManager::SmGetHeight ( )

Return the image height of video source.

Returns

size_t The image height of video source

4.5.2.8 void lambdavue::SourceManager::SmGetImage (BufferStruct & *output_frame*)

Get the decoded frame image from LambdaVue source manager.

Returns

output_frame The BufferStruct to store the frame image

4.5.2.9 std::string lambdavue::SourceManager::SmGetSourceName ( )

Get the name of video source.

Returns

std::string Current source name

4.5.2.10 size_t lambdavue::SourceManager::SmGetWidth ( )

Return the image width of video source.

Returns

size_t The image width of video source

4.5.2.11 int64_t lambdavue::SourceManager::SmQuery (enum **lambdavue::SourceAccess** *type*)

Query the ability or status of video source.

Parameters

type	Specify control type

Returns

int64_t The current value of the query type, if the type is unaccessible then UNAVAILABLE is returned

4.5.2.12 bool lambdavue::SourceManager::SmSetSource (enum SourceOperation *operation*, const char * *source*)

Video source setting or closing.

Parameters

operation	Specify the SourceOperation
source	The name of the video source (filename or device name)

The documentation for this class was generated from the following file:

- /Users/nathanchen/Desktop/LambdaVue/LambdaVueSDK/LambdaVue/source/include/source.h

4.6 lambdavue::VerificationResult Struct Reference

A struct to describe license verification result.

`#include <core.h>`

Public Attributes

- enum LicenseStatus **status**
- enum LicenseErrorCode **error_code**

4.6.1 Detailed Description

A struct to describe license verification result.

status License verification status error_code License verification error code

The documentation for this struct was generated from the following file:

- LambdaVue/core/include/core.h

4.7 lambdavue::WriterManager Class Reference

This class provides some API functions to management LambdaVue media writer WriterManager contains two MediaWriter, one for original video and the other for processed video.

`#include <writer.hpp>`

Public Member Functions

- WriterManager ()

  Constructor of writer manager.
- ~WriterManager ()

  Destructor of writer manager.
- void WmSetRecordType (enum RecordType type)

  Reccord type setting.
- void WmInitWriter (int width, int height, float framerate, int sample_rate, int channel_number)

  Initial MediaWriter.
- void WmSetOutputResolution (enum OutputFileType output_file_type, int width, int height)

  Set resolution of output video.
- void WmWriteImage (BufferStruct &img, enum RecordType)

  Encode an image buffer and write to video file.
- void WmCloseWriter (enum OutputFileType output_file_type)

  Close original and processed MediaWriter.
- void WmSetOutputDestination (enum OutputFileType output_file_type, std::string input_full_filename=string(), std::string output_file_destination=string())

  Set the destination of output videos.
- struct WriterState WmGetWriterState ()

  Return the current WriterManager status.

4.7.1 Detailed Description

This class provides some API functions to management LambdaVue media writer WriterManager contains two MediaWriter, one for original video and the other for processed video.

4.7.2 Member Function Documentation

4.7.2.1 void lambdavue::WriterManager::WmCloseWriter (enum OutputFileType *output_file_type*)

Close original and processed MediaWriter.

Parameters

output_file_type	Indicate which type of output video

4.7.2.2 struct WriterState lambdavue::WriterManager::WmGetWriterState ( )

Return the current WriterManager status.

Returns

struct WriterState The current WriterManager status

4.7.2.3 void lambdavue::WriterManager::WmInitWriter (int *width*, int *height*, float *framerate*, int *sample_rate*, int *channel_number*)

Initial MediaWriter.

Parameters

width	Width of output video
height	Height of output video
framerate	Framerate of output video
sample_rate	Audio sample rate of output video (not yet supported)
channel_number	Channel number of output video (not yet supported)

4.7.2.4 void lambdavue::WriterManager::WmSetOutputDestination (enum OutputFileType *output_file_type*, std::string *input_full_filename* = `string()`, std::string *output_file_destination* = `string()`)

Set the destination of output videos.

Parameters

output_file_type	Indicate which type of output video
input_full_filename	The filename of input video source
output_file_destination	The destination of output video

4.7.2.5 void lambdavue::WriterManager::WmSetOutputResolution (enum OutputFileType *output_file_type*, int *width*, int *height*)

Set resolution of output video.

Parameters

output_file_type	Indicate which type of output video
width	Width of output video
height	Height of output video

4.7.2.6 void lambdavue::WriterManager::WmSetRecordType (enum RecordType *type*)

Reccord type setting.

Parameters

type	Specify the record type

4.7.2.7 void lambdavue::WriterManager::WmWriteImage (BufferStruct & *img,* enum *RecordType*)

Encode an image buffer and write to video file.

Parameters

img	Image BufferStruct
RecordType	Record type

The documentation for this class was generated from the following file:

- /Users/nathanchen/Desktop/LambdaVue/LambdaVueSDK/LambdaVue/writer/include/writer.hpp

4.8 lambdavue::WriterState Struct Reference

A struct to describe current video writer status.

`#include <core.h>`

Public Attributes

- RecordType **record_type**
- int **processed_width**
- int **processed_height**
- std::string **processed_dest_filename**
- int **original_width**
- int **original_height**
- std::string **origianl_dest_filename**

4.8.1 Detailed Description

A struct to describe current video writer status.

record_type Current output video record type processed_width Width of processed video processed_height Height of processed video processed_dest_filename File name of the processed video original_width Width of original video original_height Height of original video origianl_dest_filename File name of the origianl video

The documentation for this struct was generated from the following file:

- LambdaVue/core/include/core.h

Index