



Denglin Hamming™ V2

dIJPEG API

DL-DG/SW-034B-01

2023-4-30

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

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

商标和许可



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

通知

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

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

苏州登临科技有限公司

苏州工业园区扬富路11号南岸新地一期商务楼5号1101室，江苏，中国

<http://www.denglin.ai>

Email : support@denglin.ai

Change History

Version	Change description
01	Initial version.

CONTENTS

1	Library API	3
1.1	Class Hierarchy	3
1.2	Full API	3
	Index	21

LIBRARY API**1.1 Class Hierarchy**

- *Struct DL_JPEG_COMPONENT_INFO*
- *Struct DL_JPEG_DECODE_PARAMS*
- *Struct DL_JPEG_DEVICE_INFO*
- *Struct DL_JPEG_DEVICE_PROP*
- *Struct DL_JPEG_ENCODE_PARAMS*
- *Struct DL_JPEG_HEADER_INFO*
- *Struct DL_JPEG_HUFF_TABLE*
- *Struct DL_JPEG_QUANT_TABLE*
- *Enum DL_JPEG_MIRROR_TYPE*
- *Enum DL_JPEG_PIXEL_FORMAT*
- *Enum DL_JPEG_RESULT*
- *Enum DL_JPEG_ROTATION_TYPE*
- *Enum DL_JPEG_SCALE*

1.2 Full API**1.2.1 Classes and Structs****Struct DL_JPEG_COMPONENT_INFO**

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_COMPONENT_INFO

Image component info for jpeg decode and encode

Public Members

uint8_t **hSampFactor**

Horizontal sampling factor (1..4), steps per pixel

uint8_t **vSampFactor**

Vertical sampling factor (1..4), steps per pixel

uint8_t **quantTableId**

Selects quantization table (0..3)

unsigned int **dcTableId**

Selects DC entropy table (0..3), decode only

unsigned int **acTableId**

Selects AC entropy table (0..3), decode only

Struct DL_JPEG_DECODE_PARAMS

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_DECODE_PARAMS

dJPEG decode parameters

Public Members

DL_JPEG_HEADER_INFO ***headerInfo**

Header info

DL_JPEG_SCALE **horDownScale**

Horizontal scale

DL_JPEG_SCALE **verDownScale**

Vertical scale

bool **roiEnable**

Enable ROI

uint32_t **roiOffsetX**

ROI top left x

uint32_t **roiOffsetY**

ROI top left y

uint32_t **roiWidth**

ROI area width

uint32_t **roiHeight**

ROI area height

DL_JPEG_ROTATION_TYPE **rotation**

Rotation for decode

DL_JPEG_MIRROR_TYPE **mirror**

Mirror for decode

Struct **DL_JPEG_DEVICE_INFO**

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_DEVICE_INFO

Device information, for jpeg device selection.

Public Members

uint32_t **clusterMask**

Cluster mask 0xf enables all 4 clusters

uint32_t **channelMask**

Channel mask 0xf enables all 4 channels

Struct **DL_JPEG_DEVICE_PROP**

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_DEVICE_PROP

Jpeg device properties

Public Members

uint32_t **hardwareVersion**

jpeg device hardware version

uint32_t **dliDeviceId**

dlgpu device id

uint32_t **cluster**

Cluster number

uint32_t **channel**

Channel number

uint32_t **indexInChannel**

JPU index in the channel

Struct DL_JPEG_ENCODE_PARAMS

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_ENCODE_PARAMS

dlJPEG encode parameters

Public Members

DL_JPEG_HEADER_INFO ***headerInfo**

uint16_t **quality**

Encode quality, clamped to range [1,100]

DL_JPEG_ROTATION_TYPE **rotation**

Rotation for encode

DL_JPEG_MIRROR_TYPE **mirror**

Mirror for encode

Struct DL_JPEG_HEADER_INFO

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_HEADER_INFO

Jpeg required header information for both decode and encode

Public Members

uint16_t **width**

Image width

uint16_t **height**

Image height

uint8_t **restart_interval**

Jpeg restart interval

uint8_t **numComponents**

Count of components.

DL_JPEG_COMPONENT_INFO **compInfo**[3]

Components' info.

DL_JPEG_QUANT_TABLE ***quantTables**[4]

Quantization tables. For encode, can be nullptr, and the default table is used in case of nullptr.

DL_JPEG_HUFF_TABLE ***dcHuffTables**[2]

DC huffman tables. For encode, can be nullptr, and the default table is used in case of nullptr.

DL_JPEG_HUFF_TABLE *acHuffTables[2]

AC huffman tables. For encode, can be nullptr, and the default table is used in case of nullptr.

Struct DL_JPEG_HUFF_TABLE

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_HUFF_TABLE

Jpeg Huffman table.

Public Members

uint8_t bits[16]

uint8_t huffVal[256]

Struct DL_JPEG_QUANT_TABLE

- Defined in file dljpeg.h

Struct Documentation

struct DL_JPEG_QUANT_TABLE

Jpeg quantization table

Public Members

uint8_t quantVal[64]

1.2.2 Enums

Enum DL_JPEG_MIRROR_TYPE

- Defined in file dljpeg.h

Enum Documentation

enum DL_JPEG_MIRROR_TYPE

Mirror enumerations

Values:

enumerator DL_JPEG_MIRROR_NONE = 0

No mirror

enumerator DL_JPEG_MIRROR_H = 1

Horizontal mirror

enumerator `DL_JPEG_MIRROR_V` = 2

Vertical mirror

enumerator `DL_JPEG_MIRROR_BOTH` = `DL_JPEG_MIRROR_V` | `DL_JPEG_MIRROR_H`

Both horizontal mirror and vertical mirror

Enum `DL_JPEG_PIXEL_FORMAT`

- Defined in file `dljpeg.h`

Enum Documentation

enum `DL_JPEG_PIXEL_FORMAT`

Enumeration for yuv pixel formats. Planar YUV requires 3 planes, semi-planar YUV requires 2 planes, and packed YUV requires only 1 plane.

Values:

enumerator `DL_JPEG_PIXEL_FORMAT_YUV420P` = 0

yuv420 planar

enumerator `DL_JPEG_PIXEL_FORMAT_YUV422P`

yuv422 planar

enumerator `DL_JPEG_PIXEL_FORMAT_YUV440P`

yuv440 planar

enumerator `DL_JPEG_PIXEL_FORMAT_YUV444P`

yuv444 planar

enumerator `DL_JPEG_PIXEL_FORMAT_YUV400P`

yuv400 planar

enumerator `DL_JPEG_PIXEL_FORMAT_NV12`

yuv420 semi-planar UV interleaved

enumerator `DL_JPEG_PIXEL_FORMAT_NV21`

yuv420 semi-planar VU interleaved

enumerator `DL_JPEG_PIXEL_FORMAT_NV16`

yuv422 semi-planar UV interleaved

enumerator `DL_JPEG_PIXEL_FORMAT_NV61`

yuv422 semi-planar VU interleaved

enumerator `DL_JPEG_PIXEL_FORMAT_NV24`

yuv422 semi-planar UV interleaved

enumerator `DL_JPEG_PIXEL_FORMAT_NV42`

yuv422 semi-planar VU interleaved

enumerator `DL_JPEG_PIXEL_FORMAT_YUYV422`

yuv422 packed YUYV, that is, YUY2

enumerator `DL_JPEG_PIXEL_FORMAT_UYVY422`

yuv422 packed UYVY

enumerator `DL_JPEG_PIXEL_FORMAT_YVYU422`

yuv422 packed YVYU

enumerator DL_JPEG_PIXEL_FORMAT_VYUY422
yuv422 packed VYUY

enumerator DL_JPEG_PIXEL_FORMAT_YUV444
yuv444 packed YUV

Enum DL_JPEG_RESULT

- Defined in file dljpeg.h

Enum Documentation

enum DL_JPEG_RESULT

Enumeration returned by most dJPEG APIs

Values:

enumerator DL_JPEG_RESULT_SUCCESS = 0
Success

enumerator DL_JPEG_RESULT_FAILED = -1
Generic failure

enumerator DL_JPEG_RESULT_INVALID_DEVICE = -2
Invalid jpeg device

enumerator DL_JPEG_RESULT_INSUFFICIENT_RESOURCES = -3
Insufficient resources

enumerator DL_JPEG_RESULT_OUT_OF_MEMORY = -4
Out of memory

enumerator DL_JPEG_RESULT_OUT_OF_DEVICE_MEMORY = -5
Out of device memory

enumerator DL_JPEG_RESULT_BAD_PARAMETER = -6
Bad parameter

enumerator DL_JPEG_RESULT_NOT_SUPPORTED = -7
Function not supported

enumerator DL_JPEG_RESULT_TIMEOUT = -7
Time out

enumerator DL_JPEG_RESULT_INVALID_VALUE = -8
Invalid parameter or value

enumerator DL_JPEG_RESULT_IO_ERROR = -9
Generic IO error (including fs, etc)

enumerator DL_JPEG_RESULT_CODEC_ERROR = -10
Jpeg decode or encode error

enumerator DL_JPEG_RESULT_HUFF_TABLE_ERROR = -11
Huff table or related parameter error

enumerator DL_JPEG_RESULT_QMAT_TABLE_ERROR = -12
Qmat table or related parameter error

Enum DL_JPEG_ROTATION_TYPE

- Defined in file dljpeg.h

Enum Documentation

enum DL_JPEG_ROTATION_TYPE

Rotation enumerations

Values:

enumerator DL_JPEG_ROTATE_NONE = 0

No rotation

enumerator DL_JPEG_ROTATE_90 = 1

Rotate 90 degrees, CW

enumerator DL_JPEG_ROTATE_180 = 2

Rotate 180 degrees, CW

enumerator DL_JPEG_ROTATE_270 = 3

Rotate 270 degrees, CW

Enum DL_JPEG_SCALE

- Defined in file dljpeg.h

Enum Documentation

enum DL_JPEG_SCALE

Scale enumerations. Vertical direction and horizontal direction scales are specified separately.

Values:

enumerator DL_JPEG_SCALE_NONE = 0

No scale

enumerator DL_JPEG_SCALE_ONE_HALF = 1

Scale to 1/2

enumerator DL_JPEG_SCALE_ONE_QUARTER = 2

Scale to 1/4

enumerator DL_JPEG_SCALE_ONE_EIGHTH = 3

Scale to 1/8

1.2.3 Functions

Function `dljpegCreateSession`

- Defined in file `dljpeg.h`

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegCreateSession (DL_JPEG_SESSION *session, DL_JPEG_DEVICE device)

Create a jpeg session.

Parameters

- [out] session: created a jpeg session
- [in] device: target jpeg device to create a session with

Return Value

- DL_JPEG_RESULT:

Function `dljpegDecode`

- Defined in file `dljpeg.h`

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegDecode (DL_JPEG_SESSION session, const DL_JPEG_DECODE_PARAMS

Decode jpeg synchronously.

This function will wait until hardware decode is completed.

Return DL_JPEG_RESULT

Parameters

- [in] session: jpeg session
 - [in] params: decode params
 - [in] scanline: input scanline buffer allocated by [*dljpegMalloc\(\)*](#), saves compressed jpeg data inside
 - [in] scanlineSize: *scanline* data size
 - [in] yuvPlanes: saves decoded uncompressed yuv planes. Should be allocated by [*dljpegMalloc\(\)*](#).
 - [in] yuvStrides: plane strides for *yuvPlanes*
 - [in] yuvFormat: output yuv format. For semi-planar formats, *yuvPlanes* and *yuvStrides* must specify 2 values. And for packed yuv (interleaved) formats, only 1 value is required for *yuvPlanes* and *yuvStrides*.
-

Function dljpegDecodeAsync

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegDecodeAsync (DL_JPEG_SESSION session, const DL_JPEG_DECODE_PARAMS

Decode jpeg asynchronously.

This function will return immediately before hardware decode is done. Hardware will notify *event* when decode is done.

Return DL_JPEG_RESULT

Parameters

- [in] session: jpeg session
- [in] params: decode params
- [in] scanline: input scanline buffer allocated by *dljpegMalloc()*, saves compressed jpeg data inside
- [in] scanlineSize: *scanline* data size
- [in] yuvPlanes: saves decoded uncompressed yuv planes. Should be allocated by *dljpegMalloc()*.
- [in] yuvStrides: plane strides for *yuvPlanes*
- [in] yuvFormat: output yuv format. For semi-planar formats, *yuvPlanes* and *yuvStrides* must specify 2 values. And for packed yuv (interleaved) formats, only 1 value is required for *yuvPlanes* and *yuvStrides*.
- [in] event: event that will be signaled after decode is done

Function dljpegDestroySession

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegDestroySession (DL_JPEG_SESSION session)

Destroy the jpeg session.

Parameters

- [in] session: the jpeg session to be destroyed

Return Value

- DL_JPEG_RESULT:

Function dljpegEncode

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegEncode (DL_JPEG_SESSION session, const DL_JPEG_ENCODE_PARAMS params)
Encode jpeg synchronously.

This function will wait until hardware encode is completed.

Parameters

- [in] session: jpeg session
- [in] params: encode params
- [in] yuvPlanes: saves uncompressed input yuv planes. Should be allocated by [dljpegMalloc\(\)](#).
- [in] yuvStrides: plane strides for *yuvPlanes*
- [in] yuvFormat: input yuv format. For semi-planar formats, *yuvPlanes* and *yuvStrides* must specify 2 values. And for packed yuv (interleaved) formats, only 1 value is required for *yuvPlanes* and *yuvStrides*.
- [in] scanline: output scanline buffer allocated by [dljpegMalloc\(\)](#), saves compressed jpeg data inside, MUST align to 256 byte.
- [out] scanlineSize: pointer to hold encoded *scanline* data size

Return Value

- DL_JPEG_RESULT:

Function dljpegFillFileHeader

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegFillFileHeader (void *fileHeader, size_t *actualFileHeaderSize)
Write the jpeg file header to memory.

This function writes the jpeg file header.

[dljpegEncode\(\)](#) only encodes compressed jpeg scanlines.

Parameters

- [out] fileHeader: header buffer
- [out] actualFileHeaderSize: header data size
- [in] fileHeaderSize: header buffer size
- [in] params: encode parameters

Return Value

- DL_JPEG_RESULT:

Function dljpegFree

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegFree (void *buffer)

Free the device memory for the jpeg device.

Can be replaced by cudaFree.

Parameters

- [in] buffer: device memory

Return Value

- DL_JPEG_RESULT:

Function dljpegFreeHeader

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegFreeHeader (DL_JPEG_HEADER_INFO *headerInfo)

Free allocated header info from *dljpegParseHeader()*

Return Often succeeds in normal conditions

Parameters

- [in] headerInfo: pointer header info to free

Function dljpegGetCompatibleFormats

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetCompatibleFormats (uint32_t *actualCount, DL_JPEG_PIXEL

Utility function: get compatible pixel format with header info.

Parameters

- [out] actualCount: actual compatible format count
- [inout] formats: format array to fill in compatible formats
- [in] count: format array size
- [in] headerInfo: jpeg header information

Return Value

- DL_JPEG_RESULT:
-

Function dljpegGetDevice

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetDevice (DL_JPEG_DEVICE *device, const DL_JPEG_DEVICE_INFO

Get a jpeg device via the mask.

There may be more than one jpeg device matching the mask. dlJPEG will choose one suitable jpeg device from the matched devices.

Parameters

- [out] device: hold the jpeg device matching the mask, may be nullptr if no match is found
- [in] deviceInfo: cluster mask and channel mask for device selection

Return Value

- DL_JPEG_RESULT: DL_JPEG_RESULT_SUCCESS if a matched device is found, else DL_JPEG_RESULT_FAILED

Function dljpegGetDeviceCount

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetDeviceCount (uint32_t *count, const DL_JPEG_DEVICE_INFO

Get the count of jpeg devices matching the masks.

Parameters

- [out] count: total number of the matched jpeg devices
- [in] deviceInfo: cluster mask and channel mask for device selection. If deviceInfo==nullptr, dlJPEG will get the count of all the available JPUs.

Return Value

- DL_JPEG_RESULT: DL_JPEG_RESULT_SUCCESS

Function dljpegGetDeviceProp

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetDeviceProp (DL_JPEG_DEVICE device, DL_JPEG_DEVICE_PROP prop)
Get the jpeg device property.

Parameters

- [in] device: jpeg device
- [out] prop: jpeg device property pointer

Return Value

- DL_JPEG_RESULT: DL_JPEG_RESULT_SUCCESS if the property is got.

Function dljpegGetDevices

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetDevices (uint32_t *actualCount, DL_JPEG_DEVICE *devices)
Get jpeg devices matching the masks.

If more than jpegDeviceArraySize devices match the mask, dlJPEG gets only jpegDeviceArraySize devices.

Parameters

- [out] actualCount: actual number of jpeg devices returned
- [out] devices: array to hold output devices
- [in] count: max size of *devices* array
- [in] deviceInfo: cluster mask and channel mask for device selection. If deviceInfo==nullptr, dlJPEG will get all the available JPU devices.

Return Value

- DL_JPEG_RESULT: DL_JPEG_RESULT_SUCCESS if matched devices are found, else DL_JPEG_RESULT_FAILED

Function dljpegGetImageBufferSize

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetImageBufferSize (size_t *imageBufferSize, const DL_JPEG_DEVICE device)
Get the output image buffer size for decode.

Output yuv width and height are rounded up to the multiple of 16.

Parameters

- [out] imageBufferSize: output image buffer size array

- [in] params: headerInfo initialized by *dljpegParseHeader()*, and the user has set the scale and ROI. Must be the same as the decodeParam passed to *dljpegDecode()* or *dljpegDecodeAsync()*
- [in] imageFormat: specify target image format

Return Value

- DL_JPEG_RESULT:

Function dljpegGetLastError

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetLastError (DL_JPEG_SESSION session)

Get the last error of the session.

Get the error for asynchronous decode. The error will be cleared after dljpegGetLastError.

Parameters

- [in] session:

Return Value

- DL_JPEG_RESULT:

Function dljpegGetSessionCount

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegGetSessionCount (int32_t *sessionCount, DL_JPEG_DEVICE device)

Retrieve created session count of specified device.

Parameters

- [out] sessionCount: output created session count
- [in] device: dlJpeg device

Return Value

- DL_JPEG_RESULT:

Function dljpegMalloc

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegMalloc (void **buffer, DL_JPEG_DEVICE device, size_t size)

Allocate the device memory for the jpeg device.

The jpeg device must use the device memory allocated by dljpegMalloc.

Both input and output buffers for the jpeg device require memory on the device. The input data need to be copied from the host memory to the device memory by cudaMemcpy.

The output data also can be copied from the device memory to the host memory by cudaMemcpy if needed. The memory allocated by *dljpegMalloc()* cannot be used across jpeg devices. Use cudaFree() to free memory allocated by *dljpegMalloc()*.

Parameters

- [out] buffer: device memory
- [in] device: target jpeg device
- [in] size: memory size

Return Value

- DL_JPEG_RESULT:

Function dljpegParseHeader

- Defined in file dljpeg.h

Function Documentation

DL_JPEG_RESULT DLJPEG_API dljpegParseHeader (DL_JPEG_HEADER_INFO **headerInfo, uint32_t *s

Utility function: parse the jpeg file header.

Decode only. The image width, height, format can be got from jpegInfo then.

Parameters

- [out] headerInfo: header information
- [out] sosOffset: fill in offset of the start of the scanline upon return
- [in] headerPtr: pointer to jpeg header data in host memory
- [in] headerSize: size of the jpeg header in memory

Return Value

- DL_JPEG_RESULT:

1.2.4 Typedefs

Typedef DL_JPEG_DEVICE

- Defined in file dljpeg.h

Typedef Documentation

typedef void ***DL_JPEG_DEVICE**
dJPEG device handle

Typedef DL_JPEG_SESSION

- Defined in file dljpeg.h

Typedef Documentation

typedef void ***DL_JPEG_SESSION**
dJPEG session handle

- genindex
- search

INDEX

D

- DL_JPEG_COMPONENT_INFO (C++ struct), 4
- DL_JPEG_COMPONENT_INFO::acTableId (C++ member), 4
- DL_JPEG_COMPONENT_INFO::dcTableId (C++ member), 4
- DL_JPEG_COMPONENT_INFO::hSampFactor (C++ member), 4
- DL_JPEG_COMPONENT_INFO::quantTableId (C++ member), 4
- DL_JPEG_COMPONENT_INFO::vSampFactor (C++ member), 4
- DL_JPEG_DECODE_PARAMS (C++ struct), 4
- DL_JPEG_DECODE_PARAMS::headerInfo (C++ member), 4
- DL_JPEG_DECODE_PARAMS::horDownScale (C++ member), 4
- DL_JPEG_DECODE_PARAMS::mirror (C++ member), 5
- DL_JPEG_DECODE_PARAMS::roiEnable (C++ member), 4
- DL_JPEG_DECODE_PARAMS::roiHeight (C++ member), 4
- DL_JPEG_DECODE_PARAMS::roiOffsetX (C++ member), 4
- DL_JPEG_DECODE_PARAMS::roiOffsetY (C++ member), 4
- DL_JPEG_DECODE_PARAMS::roiWidth (C++ member), 4
- DL_JPEG_DECODE_PARAMS::rotation (C++ member), 4
- DL_JPEG_DECODE_PARAMS::verDownScale (C++ member), 4
- DL_JPEG_DEVICE (C++ type), 19
- DL_JPEG_DEVICE_INFO (C++ struct), 5
- DL_JPEG_DEVICE_INFO::channelMask (C++ member), 5
- DL_JPEG_DEVICE_INFO::clusterMask (C++ member), 5
- DL_JPEG_DEVICE_PROP (C++ struct), 5
- DL_JPEG_DEVICE_PROP::channel (C++ member), 5
- DL_JPEG_DEVICE_PROP::cluster (C++ member), 5
- DL_JPEG_DEVICE_PROP::dliDeviceId (C++ member), 5
- DL_JPEG_DEVICE_PROP::hardwareVersion (C++ member), 5
- DL_JPEG_DEVICE_PROP::indexInChannel (C++ member), 5
- DL_JPEG_ENCODE_PARAMS (C++ struct), 6
- DL_JPEG_ENCODE_PARAMS::headerInfo (C++ member), 6
- DL_JPEG_ENCODE_PARAMS::mirror (C++ member), 6
- DL_JPEG_ENCODE_PARAMS::quality (C++ member), 6
- DL_JPEG_ENCODE_PARAMS::rotation (C++ member), 6
- DL_JPEG_HEADER_INFO (C++ struct), 6
- DL_JPEG_HEADER_INFO::acHuffTables (C++ member), 6
- DL_JPEG_HEADER_INFO::compInfo (C++ member), 6
- DL_JPEG_HEADER_INFO::dcHuffTables (C++ member), 6
- DL_JPEG_HEADER_INFO::height (C++ member), 6
- DL_JPEG_HEADER_INFO::numComponents (C++ member), 6
- DL_JPEG_HEADER_INFO::quantTables (C++ member), 6
- DL_JPEG_HEADER_INFO::restart_interval (C++ member), 6
- DL_JPEG_HEADER_INFO::width (C++ member), 6
- DL_JPEG_HUFF_TABLE (C++ struct), 7
- DL_JPEG_HUFF_TABLE::bits (C++ member), 7
- DL_JPEG_HUFF_TABLE::huffVal (C++ member), 7
- DL_JPEG_MIRROR_TYPE (C++ enum), 7
- DL_JPEG_MIRROR_TYPE::DL_JPEG_MIRROR_BOTH (C++ enumerator), 8
- DL_JPEG_MIRROR_TYPE::DL_JPEG_MIRROR_H (C++ enumerator), 7

