



# Denglin Hamming™ V2

## dIVID API

DL-DG/SW-033B-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

<b>1</b>	<b>Library API</b>	<b>3</b>
1.1	Class Hierarchy . . . . .	3
1.2	Full API . . . . .	4
	<b>Index</b>	<b>51</b>

**LIBRARY API****1.1 Class Hierarchy**

- *Struct DL\_VID\_BUFFER\_CREATE\_INFO*
- *Struct DL\_VID\_BUFFER\_PARAM*
- *Struct DL\_VID\_BUFFER\_PARAM\_COLOUR\_DESCRIPTION*
- *Struct DL\_VID\_BUFFER\_PARAM\_DISPLAY\_SIZE*
- *Struct DL\_VID\_DEVICE\_INFO*
- *Struct DL\_VID\_DEVICE\_PROP*
- *Struct DL\_VID\_FRAME\_ALLOC\_PARAMS*
- *Struct DL\_VID\_ROI\_REGION*
- *Struct DL\_VID\_SEQUENCE\_PARAMS*
- *Struct DL\_VID\_SESSION\_CREATE\_INFO*
- *Enum DL\_VID\_BUFFER\_DIRECTION*
- *Enum DL\_VID\_BUFFER\_FLAG*
- *Enum DL\_VID\_BUFFER\_FORMAT*
- *Enum DL\_VID\_BUFFER\_FRAME\_FLAG*
- *Enum DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG*
- *Enum DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE*
- *Enum DL\_VID\_BUFFER\_PARAM\_TYPE*
- *Enum DL\_VID\_CHROMA\_FORMAT*
- *Enum DL\_VID\_CODEC\_FORMAT*
- *Enum DL\_VID\_DEVICE\_TYPE*
- *Enum DL\_VID\_RATE\_CONTROL\_MODE*
- *Enum DL\_VID\_RESULT*
- *Enum DL\_VID\_SESSION\_STATE*

## 1.2 Full API

### 1.2.1 Classes and Structs

#### Struct DL\_VID\_BUFFER\_CREATE\_INFO

- Defined in file dlvid.h

#### Struct Documentation

**struct DL\_VID\_BUFFER\_CREATE\_INFO**

DL\_VID\_BUFFER creation info

#### Public Members

*DL\_VID\_DEVICE* **device**

The buffer is created for the dVID device.

*DL\_VID\_BUFFER\_FORMAT* **format**

Data format

uint32\_t **flags**

DL\_VID\_BUFFER\_FRAME\_FLAGS

size\_t **size**

Buffer size, *dlvidAllocateBufferMemory()* allocates the device memory size

uint32\_t **height**

Frame height, frame data only

uint32\_t **width**

Frame width, frame data only

uint32\_t **stride**

Frame data stride, frame data only

uint32\_t **strideAlign**

Frame data stride alignment, frame data only

uint32\_t **afbcAllocBytes**

afbc alloc bytes, afbc only

uint32\_t **afbcWidthInSuperblocks**

afbc width in superblocks, afbc only

#### Struct DL\_VID\_BUFFER\_PARAM

- Defined in file dlvid.h

---

## Struct Documentation

### **struct DL\_VID\_BUFFER\_PARAM**

Buffer param

#### **Public Members**

*DL\_VID\_BUFFER\_PARAM\_TYPE* **type**

Data type for union data

uint32\_t **arg**

For type DL\_VID\_BUFFER\_PARAM\_TYPE\_DPB\_HELD\_FRAMES

*DL\_VID\_BUFFER\_PARAM\_DISPLAY\_SIZE* **displaySize**

For type DL\_VID\_BUFFER\_PARAM\_TYPE\_DISPLAY\_SIZE

*DL\_VID\_BUFFER\_PARAM\_COLOUR\_DESCRIPTION* **colourDescription**

For type DL\_VID\_BUFFER\_PARAM\_TYPE\_COLOUR\_DESCRIPTION

**union** *DL\_VID\_BUFFER\_PARAM*::[anonymous] **data**

### **Struct DL\_VID\_BUFFER\_PARAM\_COLOUR\_DESCRIPTION**

- Defined in file dlvid.h

## Struct Documentation

### **struct DL\_VID\_BUFFER\_PARAM\_COLOUR\_DESCRIPTION**

Metadata needed for displaying High Dynamic Range (HDR) content

#### **Public Members**

uint32\_t **flags**

DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAGS

*DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE* **range**

Colour range

uint8\_t **colourPrimaries**

See hevc spec. E.3.1

uint8\_t **transferCharacteristics**

See hevc spec. E.3.1

uint8\_t **matrixCoeff**

See hevc spec. E.3.1

uint16\_t **masteringDisplayPrimariesX[3]**

See hevc spec. D.3.27

uint16\_t **masteringDisplayPrimariesY[3]**

See hevc spec. D.3.27

uint16\_t **masteringWhitePointX**

See hevc spec. D.3.27

uint16\_t **masteringWhitePointY**

See hevc spec. D.3.27

uint32\_t **maxDisplayMasteringLuminance**

See hevc spec. D.3.27

uint32\_t **minDisplayMasteringLuminance**

See hevc spec. D.3.27

uint32\_t **maxContentLightLevel**

Unused

uint32\_t **avgContentLightLevel**

Unused

## Struct DL\_VID\_BUFFER\_PARAM\_DISPLAY\_SIZE

- Defined in file dlvid.h

### Struct Documentation

**struct DL\_VID\_BUFFER\_PARAM\_DISPLAY\_SIZE**

Output from decoder. VP9 only

#### Public Members

uint16\_t **displayWidth**

Display width

uint16\_t **displayHeight**

Display height

## Struct DL\_VID\_DEVICE\_INFO

- Defined in file dlvid.h

### Struct Documentation

**struct DL\_VID\_DEVICE\_INFO**

dVID device info

#### Public Members

uint32\_t **clusterMask**

Cluster mask 0xf enables all 4 clusters

uint32\_t **channelMask**

Channel mask 0xf enables all 4 channels

uint32\_t **deviceTypeMask**

Device type



---

## Struct DL\_VID\_DEVICE\_PROP

- Defined in file dlvid.h

### Struct Documentation

**struct DL\_VID\_DEVICE\_PROP**

dlVID device properties

#### Public Members

uint32\_t **hardwareVersion**

dlVID device hardware version

uint32\_t **dlDeviceId**

dlgpu device id

*DL\_VID\_DEVICE\_TYPE* **deviceType**

Device type

uint32\_t **cluster**

Cluster number

uint32\_t **channel**

Channel number

uint32\_t **maxStreamCount**

Max stream count

uint32\_t **indexInChannel**

Decoder or encoder index in channel

## Struct DL\_VID\_FRAME\_ALLOC\_PARAMS

- Defined in file dlvid.h

### Struct Documentation

**struct DL\_VID\_FRAME\_ALLOC\_PARAMS**

Frame allocation info, decode only

#### Public Members

uint32\_t **planarAllocFrameHeight**

Frame height

uint32\_t **planarAllocFrameWidth**

Frame width

uint32\_t **afbcAllocBytes**

afbc frame size

uint32\_t **afbcWidthInSuperblocks**

afbc frame width in superblocks

## Struct DL\_VID\_ROI\_REGION

- Defined in file dlvid.h

### Struct Documentation

#### **struct DL\_VID\_ROI\_REGION**

Region of interest structure.

Define a region in 16x16 units. The region is macro block positions (x,y) in the range:  $mbxLeft \leq x < mbxRight$  and  $mbyTop \leq y < mbyBottom$

#### Public Members

uint16\_t **mbxLeft**

X coordinate of left macro block

uint16\_t **mbxRight**

X coordinate of right macro block

uint16\_t **mbyTop**

Y coordinate of top macro block

uint16\_t **mbyBottom**

Y coordinate of bottom macro block

int16\_t **qpDelta**

Delta relative to the default QP value

## Struct DL\_VID\_SEQUENCE\_PARAMS

- Defined in file dlvid.h

### Struct Documentation

#### **struct DL\_VID\_SEQUENCE\_PARAMS**

Frame sequence info, decode only

#### Public Members

uint32\_t **numAfbcBuffers**

Minimum number of output buffers required for afbc output

uint32\_t **numPlanarBuffers**

Minimum number of output buffers required for planar output

*DL\_VID\_CHROMA\_FORMAT* **chromaFormat**

Chroma format

uint8\_t **bitdepthChroma**

Chroma bitdepth 8 or 10

uint8\_t **bitdepthLuma**

Luma bitdepth 8 or 10

**uint8\_t interlace**  
Interlace flag 0 or 1

## Struct DL\_VID\_SESSION\_CREATE\_INFO

- Defined in file dlvid.h

## Struct Documentation

**struct DL\_VID\_SESSION\_CREATE\_INFO**  
dVID session creation info

### Public Members

**DL\_VID\_DEVICE device**  
dVID device

**DL\_VID\_CODEC\_FORMAT codecFormat**  
Codec format for this session

**void DLVIDAPI(\* pfnRpcPrint )(void \*, DL\_VID\_SESSION, const char \*)**  
Callback, dVID device requests printing info message.  
Can be ignored.

### Parameters

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- [in] const: char\* string that needs to be printed

**void DLVIDAPI(\* pfnError )(void \*, DL\_VID\_SESSION, DL\_VID\_RESULT, const char \*)**  
Callback, dVID error.  
Must reset the session and ignore callback after the error.

### Parameters

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- [in] DL\_VID\_RESULT: error code
- [in] const: char\* error string. May be nullptr.

**void DLVIDAPI(\* pfnStateChanged )(void \*, DL\_VID\_SESSION, DL\_VID\_SESSION\_STATE)**  
Callback, session state changed.

### Parameters

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- [in] DL\_VID\_SESSION\_STATE: session's new state

**void DLVIDAPI(\* pfnEmptiedInput )(void \*, DL\_VID\_SESSION, DL\_VID\_BUFFER)**

Callback, return the input buffer dVID has processed.

User should reuse the input buffer. Fill in new input data and call *dvidEmptyBuffer()* to send the buffer back to dVID.

**Parameters**

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- [in] DL\_VID\_BUFFER: returned input buffer

**void DLVIDAPI(\* pfnFilledOutput )(void \*, DL\_VID\_SESSION, DL\_VID\_BUFFER)**

Callback, return the output buffer that dVID has filled with output data.

User should reuse the output buffer after processing the output data, and send the buffer back to dVID by calling *dvidFillBuffer()*.

**Parameters**

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- [in] DL\_VID\_BUFFER: returned output buffer

**void DLVIDAPI(\* pfnInputFlushed )(void \*, DL\_VID\_SESSION)**

Callback, notify the user that the input buffers are flushed.

**Parameters**

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session

**void DLVIDAPI(\* pfnOutputFlushed )(void \*, DL\_VID\_SESSION)**

Callback, notify the user that the output buffers are flushed.

dVID session output queue is cleared, and all output buffers are returned. User should allocate output buffers when preparing buffers for decode.

**Parameters**

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session

**void DLVIDAPI(\* pfnFrameAllocParams )(void \*, DL\_VID\_SESSION, DL\_VID\_FRAME\_ALLOC\_PARAMS)**

Callback, decode only. Output frame allocation parameters.

User should save the frameAllocParams for allocating the output buffer later.

**Parameters**

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- *DL\_VID\_FRAME\_ALLOC\_PARAMS*: frame allocation parameters

---

```
void DLVIDAPI(* pfnSequenceParams ) (void *, DL_VID_SESSION, DL_VID_SEQUENCE_PARAMS *)
```

Callback, decode only. Output frame sequence parameters.

User should save the sequenceParams for allocating the output buffer later. User must call *dlvidFlushOutput()* after saving the sequenceParams.

#### Parameters

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- *DL\_VID\_SEQUENCE\_PARAMS*: frame sequence parameters

```
void DLVIDAPI(* pfnBufferParam ) (void *, DL_VID_SESSION, DL_VID_BUFFER_PARAM *)
```

Callback Buffer param. Decode only.

#### Parameters

- [in] void\*: user data pointer
- [in] DL\_VID\_SESSION: session
- *DL\_VID\_BUFFER\_PARAM*: buffer param

## 1.2.2 Enums

### Enum DL\_VID\_BUFFER\_DIRECTION

- Defined in file dlvid.h

#### Enum Documentation

**enum DL\_VID\_BUFFER\_DIRECTION**

DL\_VID\_BUFFER data direction for dVID session

*Values:*

**enumerator DL\_VID\_BUFFER\_IN = 0x0**

DL\_VID\_BUFFER is used as input

**enumerator DL\_VID\_BUFFER\_OUT = 0x1**

DL\_VID\_BUFFER is used as output

### Enum DL\_VID\_BUFFER\_FLAG

- Defined in file dlvid.h

---

## Enum Documentation

### enum DL\_VID\_BUFFER\_FLAG

Buffer flags on DL\_VID\_BUFFER

*Values:*

**enumerator DL\_VID\_BUFFER\_FLAG\_EOS = 0x1**

End of stream

**enumerator DL\_VID\_BUFFER\_FLAG\_STARTTIME = 0x2**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_DECODEONLY = 0x4**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_DATACORRUPT = 0x8**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_ENDOFFRAME = 0x10**

End of frame. Recommend set this flag on input buffer when input data is one or more whole frame data. Especially when using timestamp on input buffer.

**enumerator DL\_VID\_BUFFER\_FLAG\_SYNCFRAME = 0x20**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_EXTRADATA = 0x40**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_CODECCONFIG = 0x80**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_TIMESTAMPINVALID = 0x100**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_READONLY = 0x200**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_NDOFSUBFRAME = 0x400**

Reserved

**enumerator DL\_VID\_BUFFER\_FLAG\_SKIPFRAME = 0x800**

Reserved

### Enum DL\_VID\_BUFFER\_FORMAT

- Defined in file dlvid.h

## Enum Documentation

### enum DL\_VID\_BUFFER\_FORMAT

Data format in DL\_VID\_BUFFER

*Values:*

**enumerator DL\_VID\_BUFFER\_FORMAT\_BITSTREAM = 0**

Compressed bitstream data, such as h.264, h.265

**enumerator DL\_VID\_BUFFER\_FORMAT\_YUV420P = 1**

YUV420 planar (3 planes)

**enumerator DL\_VID\_BUFFER\_FORMAT\_YUV420\_SEMIPLANAR = 2**  
Semiplanar YUV (2 planes)

**enumerator DL\_VID\_BUFFER\_FORMAT\_YVU420\_SEMIPLANAR = 3**  
Semiplanar YVU (2 planes)

**enumerator DL\_VID\_BUFFER\_FORMAT\_YUYVY\_10B = 4**  
ARM 10-bit YUV 420 format

**enumerator DL\_VID\_BUFFER\_FORMAT\_YUV420\_AFBC = 5**  
YUV420 compressed with arm AFBC

**enumerator DL\_VID\_BUFFER\_FORMAT\_YUV420\_AFBC\_10B = 6**  
10-bit YUV buffer compressed with AFBC

**enumerator DL\_VID\_BUFFER\_FORMAT\_YUV422\_1P = 7**  
YUV 422 buffer (1 plane, YUY2)

**enumerator DL\_VID\_BUFFER\_FORMAT\_YVU422\_1P = 8**  
YVU 422 buffer (1 plane, UYVY)

**enumerator DL\_VID\_BUFFER\_FORMAT\_YV12 = 9**  
Planar YV12 buffer (3 planes)

**enumerator DL\_VID\_BUFFER\_FORMAT\_RGBA\_8888 = 10**  
RGB format with 32 bit as Red 31:24, Green 23:16, Blue 15:8, Alpha 7:0

**enumerator DL\_VID\_BUFFER\_FORMAT\_BGRA\_8888 = 11**  
RGB format with 32 bit as Blue 31:24, Green 23:16, Red 15:8, Alpha 7:0

**enumerator DL\_VID\_BUFFER\_FORMAT\_ARGB\_8888 = 12**  
RGB format with 32 bit as Alpha 31:24, Red 23:16, Green 15:8, Blue 7:0

**enumerator DL\_VID\_BUFFER\_FORMAT\_ABGR\_8888 = 13**  
RGB format with 32 bit as Alpha 31:24, Blue 23:16, Green 15:8, Red 7:0

**enumerator DL\_VID\_BUFFER\_FORMAT\_YUV420\_I420\_10 = 14**  
YUV420 16 bit per component (3 planes)

## Enum DL\_VID\_BUFFER\_FRAME\_FLAG

- Defined in file dlvid.h

## Enum Documentation

**enum DL\_VID\_BUFFER\_FRAME\_FLAG**

Frame flags on DL\_VID\_BUFFER

*Values:*

**enumerator DL\_VID\_BUFFER\_FRAME\_FLAG\_SCALING\_2 = 0x1**  
Decode only. Frame is scaled by half.

**enumerator DL\_VID\_BUFFER\_FRAME\_FLAG\_SCALING\_4 = 0x2**  
Decode only. Frame is scaled by a quarter.

**enumerator DL\_VID\_BUFFER\_FRAME\_FLAG\_SCALING\_MASK = 0x3**  
Scaling mask

**enumerator DL\_VID\_BUFFER\_FRAME\_FLAG\_ROTATION\_90 = 0x10**  
Frame is rotated 90 degrees

**enumerator DL\_VID\_BUFFER\_FRAME\_FLAG\_ROTATION\_180** = 0x20

Frame is rotated 180 degrees

**enumerator DL\_VID\_BUFFER\_FRAME\_FLAG\_ROTATION\_270** = 0x30

Frame is rotated 270 degrees

**enumerator DL\_VID\_BUFFER\_FRAME\_FLAG\_ROTATION\_MASK** = 0x30

Rotation mask

## Enum DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG

- Defined in file dlvid.h

### Enum Documentation

**enum DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG**

Buffer param colour flag

*Values:*

**enumerator DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG\_MASTERING\_DISPLAY\_DATA\_VALID** = 1

**enumerator DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG\_CONTENT\_LIGHT\_DATA\_VALID** = 2

## Enum DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE

- Defined in file dlvid.h

### Enum Documentation

**enum DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE**

Buffer param colour range

*Values:*

**enumerator DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE\_UNSPECIFIED** = 0

**enumerator DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE\_LIMITED** = 1

**enumerator DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE\_FULL** = 2

## Enum DL\_VID\_BUFFER\_PARAM\_TYPE

- Defined in file dlvid.h



## Enum Documentation

### enum DL\_VID\_BUFFER\_PARAM\_TYPE

Buffer param type

*Values:*

**enumerator DL\_VID\_BUFFER\_PARAM\_TYPE\_DISPLAY\_SIZE = 1**

Display size

**enumerator DL\_VID\_BUFFER\_PARAM\_TYPE\_COLOUR\_DESCRIPTION = 2**

Colour description for HDR

**enumerator DL\_VID\_BUFFER\_PARAM\_TYPE\_DPB\_HELD\_FRAMES = 3**

Frame held in the decoded picture buffer. Decode only

### Enum DL\_VID\_CHROMA\_FORMAT

- Defined in file dlvid.h

## Enum Documentation

### enum DL\_VID\_CHROMA\_FORMAT

*Values:*

**enumerator DL\_VID\_CHROMA\_FORMAT\_MONO = 0**

**enumerator DL\_VID\_CHROMA\_FORMAT\_420 = 1**

**enumerator DL\_VID\_CHROMA\_FORMAT\_422 = 2**

**enumerator DL\_VID\_CHROMA\_FORMAT\_440 = 3**

**enumerator DL\_VID\_CHROMA\_FORMAT\_ARGB = 4**

### Enum DL\_VID\_CODEC\_FORMAT

- Defined in file dlvid.h

## Enum Documentation

### enum DL\_VID\_CODEC\_FORMAT

dVID codec format

*Values:*

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_H264 = 0**

Decode h264

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_H263**

Decode h263

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_MPEG4**

Decode mpeg4

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_MPEG2**

Decode mpeg2

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_VC1**  
Decode VC1

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_VP8**  
Decode vp8

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_H265**  
Decode h265

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_VP9**  
Decode vp9

**enumerator DL\_VID\_CODEC\_FORMAT\_DECODE\_AVS**  
Decode AVS

**enumerator DL\_VID\_CODEC\_FORMAT\_ENCODE\_H264**  
Encode h264

**enumerator DL\_VID\_CODEC\_FORMAT\_ENCODE\_H265**  
Encode h265

**enumerator DL\_VID\_CODEC\_FORMAT\_INVALID**  
Invalid codec format

## Enum DL\_VID\_DEVICE\_TYPE

- Defined in file dlvid.h

## Enum Documentation

**enum DL\_VID\_DEVICE\_TYPE**  
dVID device types

*Values:*

**enumerator DL\_VID\_DEVICE\_DECODER** = 0x1  
Decoder

**enumerator DL\_VID\_DEVICE\_ENCODER** = 0x2  
Encoder

## Enum DL\_VID\_RATE\_CONTROL\_MODE

- Defined in file dlvid.h

## Enum Documentation

**enum DL\_VID\_RATE\_CONTROL\_MODE**  
Rate control mode

*Values:*

**enumerator DL\_VID\_RATE\_CONTROL\_MODE\_OFF** = 0  
Rate control off. Fixed qp mode. Default mode. The target bitrate value is ignored and the qp values are used. If the qp values are not set, the default built-in values are used.

---

**enumerator DL\_VID\_RATE\_CONTROL\_MODE\_STANDARD = 1**

Standard mode targets to match bitrate target\_bitrate when averaged over a GOP cycle. Within a GOP cycle, frames target the same quality, so an I frame is typically a few times larger than the size of a P or B frame.

**enumerator DL\_VID\_RATE\_CONTROL\_MODE\_VARIABLE = 2**

Variable mode

**enumerator DL\_VID\_RATE\_CONTROL\_MODE\_CONSTANT = 3**

Constant mode

## Enum DL\_VID\_RESULT

- Defined in file dlvid.h

## Enum Documentation

**enum DL\_VID\_RESULT**

dVID result values

*Values:*

**enumerator DL\_VID\_RESULT\_SUCCESS = 0**

Success

**enumerator DL\_VID\_RESULT\_FAILED = -1**

Failed

**enumerator DL\_VID\_RESULT\_INVALID\_DEVICE = -2**

Invalid dVID device

**enumerator DL\_VID\_RESULT\_UNDEFINED\_ERROR = -3**

Undefined error

**enumerator DL\_VID\_RESULT\_BAD\_PARAMETER = -4**

Bad function parameter

**enumerator DL\_VID\_RESULT\_BAD\_BUFFER\_DIRECTION = -5**

Bad buffer direction

**enumerator DL\_VID\_RESULT\_HARDWARE\_INIT\_ERROR = -6**

dVID device hardware initiation error

**enumerator DL\_VID\_RESULT\_HARDWARE\_ERROR = -7**

dVID device hardware error

**enumerator DL\_VID\_RESULT\_INSUFFICIENT\_RESOURCES = -8**

Insufficient resources

**enumerator DL\_VID\_RESULT\_NOT\_IMPLEMENTED = -9**

Function not implemented

**enumerator DL\_VID\_RESULT\_NOT\_READY = -10**

dVID device not ready

**enumerator DL\_VID\_RESULT\_TIMEOUT = -11**

Timeout

**enumerator DL\_VID\_RESULT\_VERSION\_MISMATCH = -12**

Hardware and driver version mismatch

---

**enumerator** `DL_VID_RESULT_STREAM_CORRUPT` = -13  
Bitstream corrupt

**enumerator** `DL_VID_RESULT_STREAM_NOT_SUPPORTED` = -14  
Bitstream format not supported

**enumerator** `DL_VID_RESULT_SESSION_HANG` = -15  
Session hang

**enumerator** `DL_VID_RESULT_IO_ERROR` = -16  
IO error

**enumerator** `DL_VID_RESULT_SESSION_LIMIT` = -100  
Session limit

### Enum `DL_VID_SESSION_STATE`

- Defined in file `dlvid.h`

### Enum Documentation

**enum** `DL_VID_SESSION_STATE`  
Session state

*Values:*

**enumerator** `DL_VID_SESSION_STATE_STOPPED` = 0  
Session stopped

**enumerator** `DL_VID_SESSION_STATE_RUNNING` = 2  
Session is running

**enumerator** `DL_VID_SESSION_STATE_PENDING` = 4  
Requested for hardware state change and waiting for the response

## 1.2.3 Functions

### Function `dlvidActivateSession`

- Defined in file `dlvid.h`

### Function Documentation

**`DL_VID_RESULT DLVID_API dlvidActivateSession (DL_VID_SESSION session)`**  
Activate the dVID session.

Must activate the session before calling *`dlvidRunSession()`*.

#### Parameters

- [in] `session`: dVID session

#### Return Value

- `DL_VID_RESULT`:

## Function dlvidAfbcDecode

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidAfbcDecode** (uint32\_t bufferCount, DL\_VID\_BUFFER \*inputBuffers,

Deprecated. Decode afbc buffers.

Decode 1~4 afbc buffers concurrently. Buffers must be in different channels, and buffers can be nullptr if the count of afbc buffers is less than 4. This function is asynchronous. Use cudaDeviceSynchronize() or cudaStreamSynchronize() to synchronize.

### Parameters

- [in] bufferCount: count of afbc buffers
- [in] inputBuffers: buffer array with afbc data, and afbc data must be ready before the user calls this function
- [out] outputs: output device memory array

### Return Value

- DL\_VID\_RESULT:

## Function dlvidAllocateBufferMemory

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidAllocateBufferMemory** (DL\_VID\_BUFFER buffer)

Allocate device memory for DL\_VID\_BUFFER.

The allocated size is the size in [\*DL\\_VID\\_BUFFER\\_CREATE\\_INFO\*](#) when creating the buffer.

### Parameters

- [in] buffer: DL\_VID\_BUFFER

### Return Value

- DL\_VID\_RESULT:

## Function dlvidCreateBuffer

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidCreateBuffer** (const DL\_VID\_BUFFER\_CREATE\_INFO \*bufferCreateInfo)  
Create DL\_VID\_BUFFER.

DL\_VID\_BUFFER is used as input and output buffers for dVID. Use *dlvidAllocateBufferMemory()* to allocate memory for buffers.

### Parameters

- [in] bufferCreateInfo: buffer creation info
- [out] buffer: created DL\_VID\_BUFFER

### Return Value

- DL\_VID\_RESULT:

## Function dlvidCreateSession

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidCreateSession** (DL\_VID\_SESSION \*session, const DL\_VID\_SESSION\_CREATE\_INFO \*sessionCreateInfo)  
Create a dVID session.

### Parameters

- [out] session: created dVID session
- [in] sessionCreateInfo: session creation info

### Return Value

- DL\_VID\_RESULT:

## Function dlvidDestroyBuffer

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidDestroyBuffer** (DL\_VID\_BUFFER buffer)  
Destroy DL\_VID\_BUFFER.

### Parameters

- [in] buffer: DL\_VID\_BUFFER

### Return Value

- DL\_VID\_RESULT:
-

## Function dlvidDestroySession

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidDestroySession (DL\_VID\_SESSION session)**

Destroy the dVID session.

### Parameters

- [in] session: dVID session to be destroyed

### Return Value

- DL\_VID\_RESULT:

## Function dlvidEmptyBuffer

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidEmptyBuffer (DL\_VID\_SESSION session, DL\_VID\_BUFFER buffer)**

Send the buffer to the session as the input buffer.

### Parameters

- [in] session:
- [in] buffer:

### Return Value

- DL\_VID\_RESULT:

## Function dlvidFillBuffer

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidFillBuffer (DL\_VID\_SESSION session, DL\_VID\_BUFFER buffer)**

Send the buffer to the session as the output buffer.

### Parameters

- [in] session:
- [in] buffer:

### Return Value

- DL\_VID\_RESULT:
-

## Function dlvidFlushInput

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidFlushInput (DL\_VID\_SESSION session)**

Flush input buffers in the session.

Note this API is only allowed when session is stopped. Othrewise may cause session error and stop work.

Note: This API is only allowed after the session is stopped, otherwise it may cause session error and the session may stop working.

### Parameters

- [in] session:

### Return Value

- DL\_VID\_RESULT:

## Function dlvidFlushOutput

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidFlushOutput (DL\_VID\_SESSION session)**

Flush output buffers in the session.

For decode, must flush the output buffer before sending the output buffer to the session after pfnSequenceParams() callback. For both decode and encode, the output buffer needs to be flushed after stopping the session at the end of codec.

Note this API is only allowed after pfnSequenceParams() callback or when session is stopped. Othrewise may cause session error and stop work.

Note: This API is only allowed after pfnSequenceParams() callback or after the session is stopped, otherwise it may cause session error and the session may stop working.

### Parameters

- [in] session:

### Return Value

- DL\_VID\_RESULT:



## Function dlvidFreeBufferMemory

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidFreeBufferMemory (DL\_VID\_BUFFER buffer)**

Free device memory for DL\_VID\_BUFFER.

### Parameters

- [in] buffer: DL\_VID\_BUFFER

### Return Value

- DL\_VID\_RESULT:

## Function dlvidGetBufferFilledLength

- Defined in file dlvid.h

## Function Documentation

**size\_t DLVID\_API dlvidGetBufferFilledLength (DL\_VID\_BUFFER buffer)**

Get the valid data size of the buffer.

Used with output buffers.

### Parameters

- [in] buffer: DL\_VID\_BUFFER

### Return Value

- size\_t: valid data size of the buffer

## Function dlvidGetBufferFlags

- Defined in file dlvid.h

## Function Documentation

**uint32\_t DLVID\_API dlvidGetBufferFlags (DL\_VID\_BUFFER buffer)**

Get the DL\_VID\_BUFFER\_FLAGS on the buffer.

### Parameters

- [in] buffer: DL\_VID\_BUFFER

### Return Value

- DL\_VID\_BUFFER\_FLAGS: on the buffer

## Function dlvidGetBufferMemoryPointer

- Defined in file dlvid.h

### Function Documentation

**DLVID\_API void \* dlvidGetBufferMemoryPointer (DL\_VID\_BUFFER buffer)**

Get the CUDA memory pointer of the buffer.

The pointer can be passed to cudaMemcpy, etc.

#### Parameters

- [in] buffer:

#### Return Value

- void\*: CUDA memory pointer of the buffer

## Function dlvidGetBufferTimestamp

- Defined in file dlvid.h

### Function Documentation

**uint64\_t DLVID\_API dlvidGetBufferTimestamp (DL\_VID\_BUFFER buffer)**

Get timestamp on the buffer.

#### Parameters

- [in] buffer: DL\_VID\_BUFFER

#### Return Value

- uint64\_t: timestamp on the buffer

## Function dlvidGetDevice

- Defined in file dlvid.h

### Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidGetDevice (DL\_VID\_DEVICE \*vidDevice, const DL\_VID\_DEVICE\_INFO**

Get a dVID device via the masks and the type.

There may be more than one dVID device matching the masks and the type. dVID will choose one suitable dVID device from the matched devices.

#### Parameters

- [out] vidDevice: dVID device matching the masks and the type, may be nullptr if no matched dVID device is found
  - [in] vidDeviceInfo: cluster mask, channel mask, and device type.
-

**Return Value**

- `DL_VID_RESULT`: `DL_VID_RESULT_SUCCESS` if a matched device is found, else `DL_VID_RESULT_FAILED`

**Function `dlvidGetDeviceCount`**

- Defined in file `dlvid.h`

**Function Documentation**

**`DL_VID_RESULT DLVID_API dlvidGetDeviceCount (int32_t *vidDeviceCount, const DL_VID_DEVICE_`**  
Get the count of dVID devices matching the masks and the type.

**Parameters**

- [out] `vidDeviceCount`: the count of matched dVID devices
- [in] `vidDeviceInfo`: cluster mask, channel mask, and device type. If `vidDeviceInfo` is nullptr, dVID will get the count of all the available decoder devices and encoder devices.

**Return Value**

- `DL_VID_RESULT`: `DL_VID_RESULT_SUCCESS`

**Function `dlvidGetDeviceProp`**

- Defined in file `dlvid.h`

**Function Documentation**

**`DL_VID_RESULT DLVID_API dlvidGetDeviceProp (DL_VID_DEVICE vidDevice, DL_VID_DEVICE_PROP *v`**  
Get the dVID device property.

**Parameters**

- [in] `vidDevice`: dVID device
- [out] `vidDeviceProp`: dVID device property pointer

**Return Value**

- `DL_VID_RESULT`: `DL_VID_RESULT_SUCCESS` if the property is got

**Function `dlvidGetDevices`**

- Defined in file `dlvid.h`

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidGetDevices** (int32\_t \*actualVidDeviceCount, DL\_VID\_DEVICE \*vidDeviceArray, DL\_VID\_DEVICE \*vidDeviceInfo)  
Get dVID devices matching the masks and the type.

If more than vidDeviceArraySize devices matching the masks and the type, dVID will get only vidDeviceArraySize devices.

### Parameters

- [out] actualVidDeviceCount: count of got dVID devices
- [out] vidDevices: DL\_VID\_DEVICE array
- [in] vidDeviceCount: dVID device array size
- [in] vidDeviceInfo: cluster mask, channel mask, and device type. If vidDeviceInfo is nullptr, dVID will get all the available decoder devices and encoder devices.

### Return Value

- DL\_VID\_RESULT: DL\_VID\_RESULT\_SUCCESS if matched devices are found, else DL\_VID\_RESULT\_FAILED

## Function dlvidGetSessionCount

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidGetSessionCount** (int32\_t \*sessionCount, DL\_VID\_DEVICE vidDevice)  
Retrieve created session count of specified device.

### Parameters

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

### Return Value

- DL\_VID\_RESULT:

## Function dlvidGetSessionDevice

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_DEVICE DLVID\_API dlvidGetSessionDevice (DL\_VID\_SESSION session)**

Get the dVID device via the session.

### Parameters

- [in] session: dVID session

### Return Value

- DL\_VID\_DEVICE: session associated dVID device

## Function dlvidIsCodecFormatSupported

- Defined in file dlvid.h

## Function Documentation

**bool DLVID\_API dlvidIsCodecFormatSupported (DL\_VID\_DEVICE vidDevice, DL\_VID\_CODEC\_FORMAT codecFormat)**

Judge if the codec format is supported by the dVID device or not.

### Parameters

- [in] vidDevice: dVID device
- [in] codecFormat: codec format

### Return Value

- bool: True if the codec format is supported by the dVID device, otherwise false.

## Function dlvidIsRgbInputSupported

- Defined in file dlvid.h

## Function Documentation

**bool DLVID\_API dlvidIsRgbInputSupported (DL\_VID\_DEVICE vidDevice)**

Judge if the RGB input is supported by the dVID encode device or not.

### Parameters

- [in] vidDevice: dVID device

### Return Value

- bool: True if the RGB input is supported by the dVID encode device, otherwise false.

## Function dlvidProcessSessionEvent

- Defined in file dlvid.h

### Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidProcessSessionEvent** (DL\_VID\_SESSION session, void \*userData, void \*timeout)  
Process the event from the session.

User must call this function repeatedly to process events from the session when the session is running.

#### Parameters

- [in] session: session that will use the buffer
- [in] userData: user data pointer to call callbacks
- [in] timeout: timeout in ms if no event is from the session

#### Return Value

- DL\_VID\_RESULT: DL\_VID\_RESULT\_TIMEOUT if timeout

## Function dlvidRegisterBuffer

- Defined in file dlvid.h

### Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidRegisterBuffer** (DL\_VID\_SESSION session, DL\_VID\_BUFFER\_DIRECTION direction, void \*buffer)  
Register DL\_VID\_BUFFER to the session.

DL\_VID\_BUFFER must be registered before being sent to the dVID session by *dlvidEmptyBuffer()* or *dlvidFillBuffer()*. Do not register a buffer more than once.

#### Parameters

- [in] session: session that will use the buffer
- [in] direction: registered as input or output buffer
- [in] buffer: buffer to be registered

#### Return Value

- DL\_VID\_RESULT:

## Function dlvidRunSession

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidRunSession (DL\_VID\_SESSION session)**

Instruct the dVID device to run the session.

The dVID device starts working on this session when the session is running. The callback pfnStateChanged will be called with DL\_VID\_SESSION\_STATE\_RUNNING.

### Parameters

- [in] session: dVID session

### Return Value

- DL\_VID\_RESULT:

## Function dlvidSetBitrate

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetBitrate (DL\_VID\_SESSION session, DL\_VID\_RATE\_CONTROL\_MODE mode, int bitrate)**

Set the rate control mode and bitrate for encode.

Encode only. Must be called after activating the session.

### Parameters

- [in] session:
- [in] mode: rate control mode
- [in] bitrate: in bits per second, taking no effect if the mode is set to DL\_VID\_RATE\_CONTROL\_MODE\_OFF

## Function dlvidSetBufferFilledLength

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetBufferFilledLength (DL\_VID\_BUFFER buffer, size\_t size)**

Set the valid data size of the buffer.

Used with input buffers.

### Parameters

- [in] buffer: DL\_VID\_BUFFER
-

- [in] size: valid data size

#### Return Value

- DL\_VID\_RESULT:

### Function dlvidSetBufferFlags

- Defined in file dlvid.h

#### Function Documentation

**void DLVID\_API dlvidSetBufferFlags (DL\_VID\_BUFFER buffer, uint32\_t flags)**  
Set the DL\_VID\_BUFFER\_FLAGS on the buffer.

##### Parameters

- [in] buffer: DL\_VID\_BUFFER
- [in] flags: combine of DL\_VID\_BUFFER\_FLAG

### Function dlvidSetBufferMemoryPointer

- Defined in file dlvid.h

#### Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetBufferMemoryPointer (DL\_VID\_BUFFER buffer, void \*memory)**  
Set CUDA memory pointer of the buffer.

Alternative way to specify underlying memory of the buffer. Instead of allocating memory internally by *dlvidAllocateBufferMemory()*, this function sets associated memory to externally allocated cuda memory pointer.

Caller should handle external cuda memory life scope. And do not call *dlvidFreeBufferMemory()* on such a buffer.

The memory must be 4096 alignment channel memory for dlvid.

##### Parameters

- [in] buffer:
- [in] memory: the memory must be channel memory and the size is equal to or larger than the size in *DL\_VID\_BUFFER\_CREATE\_INFO*

##### Return Value

- void\*: CUDA memory pointer of buffer



## Function dlvidSetBufferRoiRegions

- Defined in file dlvid.h

### Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetBufferRoiRegions (DL\_VID\_BUFFER buffer, uint32\_t regionCount)**  
Set buffer ROI regions.

Encode only. This feature enables the user to directly set the quantization parameter differently in different rectangular regions of each frame. The dlvidSetQp sets the base QP to be used, and the struct [DL\\_VID\\_ROI\\_REGION](#) then maps out rectangular regions where a QP offset is applied from the base value. Up to 16 different regions can be used.

The region-of-interest feature cannot be combined with automatic rate control, the rate control mode must be DL\_VID\_RATE\_CONTROL\_MODE\_OFF which is the default mode.

When regions overlap, the lowest indexed region in the list takes priority. If a region goes outside the frame, then parts of the region outside the frame are ignored. In effect, the regions are clipped to the frame size.

To remove ROI regions on a buffer, call this API with regionCount=0.

#### Parameters

- [in] buffer:
- [in] regionCount: region array size. Max size DL\_VID\_ROI\_COUNT\_MAX.
- [in] regions: region array

#### Return Value

- DL\_VID\_RESULT:

## Function dlvidSetBufferTimestamp

- Defined in file dlvid.h

### Function Documentation

**void DLVID\_API dlvidSetBufferTimestamp (DL\_VID\_BUFFER buffer, uint64\_t timestamp)**  
Set timestamp on the buffer.

#### Parameters

- [in] buffer: DL\_VID\_BUFFER
- [in] timestamp:

## Function dlvidSetFrameRate

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetFrameRate (DL\_VID\_SESSION session, float fps)**

Set the frame rate for encode.

Encode only. Must be called after activating the session.

### Parameters

- [in] session:
- [in] fps: frame per second

## Function dlvidSetProfileLevel

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetProfileLevel (DL\_VID\_SESSION session, uint16\_t profile, ui**

Set the profile and level for encode.

Encode only. Must be called after activating the session. In general, this function needs not to be called. An appropriate value is selected automatically.

### Parameters

- [in] session:
- [in] profile: see the profile macro definition
- [in] level: see the level macro definition

## Function dlvidSetQp

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetQp (DL\_VID\_SESSION session, uint32\_t qp)**

Set the quantization parameter for encode.

Encode only. Must be called after activating the session. Only take effect in DL\_VID\_RATE\_CONTROL\_MODE\_OFF mode which is the default mode. This is the simple version of *dlvidSetQpIPB()*. The valid ranges are: H264: 0-51, HEVC: 0-51.

### Parameters

- [in] session:
  - [in] qp: quantization parameter
-

## Function dlvidSetQpIPB

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidSetQpIPB (DL\_VID\_SESSION session, uint32\_t qp\_i, uint32\_t qp\_p,**

Set quantization parameters for I, P, B frames separately.

Encode only. Must be called after activating the session. See [\*dlvidSetQp\(\)\*](#)

### Parameters

- [in] session:
- [in] qp\_i: quantization parameter for i frame
- [in] qp\_p: quantization parameter for p frame
- [in] qp\_b: quantization parameter for b frame

## Function dlvidStopSession

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidStopSession (DL\_VID\_SESSION session)**

Instruct the dVID device to stop the session.

The callback pfnStateChanged will be called with DL\_VID\_SESSION\_STATE\_STOPPED when the session is stopped.

### Parameters

- [in] session: dVID session

### Return Value

- DL\_VID\_RESULT:

## Function dlvidStreamAfbcdDecode

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidStreamAfbcdDecode** (uint32\_t bufferCount, DL\_VID\_BUFFER \*inputB

Deprecated. Decode afbc buffers by using the given stream.

Same as *dlvidAfbcdDecode()* and the user can synchronize the decode on the given stream.

### Parameters

- [in] bufferCount: count of afbc buffers
- [in] inputBuffers: buffer array with afbc data, and afbc data must be ready before the user calls this function
- [out] outputs: output device memory array
- [in] stream: CUDA stream

### Return Value

- DL\_VID\_RESULT:

## Function dlvidUnregisterBuffer

- Defined in file dlvid.h

## Function Documentation

**DL\_VID\_RESULT DLVID\_API dlvidUnregisterBuffer** (DL\_VID\_SESSION session, DL\_VID\_BUFFER buffer

Unregister DL\_VID\_BUFFER from the session.

Unregister the buffer when the buffer does not need to be sent to the session any more, for example, end of decode.

### Parameters

- [in] session: the buffer is unregistered from the session
- [in] buffer: buffer to be unregistered

### Return Value

- DL\_VID\_RESULT:

## 1.2.4 Defines

### Define DL\_VID\_BUFFER\_ALIGNMENT

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_BUFFER\_ALIGNMENT**

## Define DL\_VID\_LEVEL\_H264\_1

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_1**

## Define DL\_VID\_LEVEL\_H264\_11

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_11**

## Define DL\_VID\_LEVEL\_H264\_12

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_12**

## Define DL\_VID\_LEVEL\_H264\_13

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_13**

## Define DL\_VID\_LEVEL\_H264\_1b

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_1b**

### Define DL\_VID\_LEVEL\_H264\_2

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_2**

### Define DL\_VID\_LEVEL\_H264\_21

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_21**

### Define DL\_VID\_LEVEL\_H264\_22

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_22**

### Define DL\_VID\_LEVEL\_H264\_3

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_3**

### Define DL\_VID\_LEVEL\_H264\_31

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_31**

### Define DL\_VID\_LEVEL\_H264\_32

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_32**

### Define DL\_VID\_LEVEL\_H264\_4

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_4**

### Define DL\_VID\_LEVEL\_H264\_41

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_41**

### Define DL\_VID\_LEVEL\_H264\_42

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_42**

### Define DL\_VID\_LEVEL\_H264\_5

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_5**

### Define DL\_VID\_LEVEL\_H264\_51

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_51**

### Define DL\_VID\_LEVEL\_H264\_52

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_52**

### Define DL\_VID\_LEVEL\_H264\_6

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_6**

### Define DL\_VID\_LEVEL\_H264\_61

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H264\_61**

### Define DL\_VID\_LEVEL\_H264\_62

- Defined in file dlvid.h



## Define Documentation

**DL\_VID\_LEVEL\_H264\_62**

## Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_1

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_1**

## Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_2

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_2**

## Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_21

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_21**

## Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_3

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_3**

## Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_31

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_31**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_4

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_4**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_41

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_41**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_5

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_5**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_51

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_51**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_52

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_52**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_6

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_6**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_61

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_61**

### Define DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_62

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_HIGH\_TIER\_62**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_1

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_1**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_2

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_2**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_21

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_21**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_3

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_3**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_31

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_31**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_4

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_4**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_41

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_41**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_5

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_5**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_51

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_51**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_52

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_52**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_6

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_6**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_61

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_61**

### Define DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_62

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_LEVEL\_H265\_MAIN\_TIER\_62**

### Define DL\_VID\_PROFILE\_H264\_BASELINE

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_PROFILE\_H264\_BASELINE**

### Define DL\_VID\_PROFILE\_H264\_HIGH

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_PROFILE\_H264\_HIGH**

### Define DL\_VID\_PROFILE\_H264\_HIGH\_10

- Defined in file dlvid.h

### Define Documentation

**DL\_VID\_PROFILE\_H264\_HIGH\_10**

### Define DL\_VID\_PROFILE\_H264\_MAIN

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_PROFILE\_H264\_MAIN**

## Define DL\_VID\_PROFILE\_H265\_MAIN

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_PROFILE\_H265\_MAIN**

## Define DL\_VID\_PROFILE\_H265\_MAIN\_10

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_PROFILE\_H265\_MAIN\_10**

## Define DL\_VID\_PROFILE\_H265\_MAIN\_INTRA

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_PROFILE\_H265\_MAIN\_INTRA**

## Define DL\_VID\_PROFILE\_H265\_MAIN\_STILL

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_PROFILE\_H265\_MAIN\_STILL**

## Define DL\_VID\_ROI\_COUNT\_MAX

- Defined in file dlvid.h

## Define Documentation

**DL\_VID\_ROI\_COUNT\_MAX**

## Define DLVIDAPI

- Defined in file dlvid.h

## Define Documentation

**DLVIDAPI**

## 1.2.5 Typedefs

### Typedef DL\_VID\_BUFFER

- Defined in file dlvid.h

### Typedef Documentation

**typedef** void \***DL\_VID\_BUFFER**  
dVID buffer handle

### Typedef DL\_VID\_BUFFER\_DIRECTION

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_BUFFER\_DIRECTION* **DL\_VID\_BUFFER\_DIRECTION**  
DL\_VID\_BUFFER data direction for dVID session

### Typedef DL\_VID\_BUFFER\_FLAG

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_BUFFER\_FLAG* **DL\_VID\_BUFFER\_FLAG**  
Buffer flags on DL\_VID\_BUFFER



### Typedef DL\_VID\_BUFFER\_FORMAT

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_BUFFER\_FORMAT* **DL\_VID\_BUFFER\_FORMAT**  
Data format in DL\_VID\_BUFFER

### Typedef DL\_VID\_BUFFER\_FRAME\_FLAG

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_BUFFER\_FRAME\_FLAG* **DL\_VID\_BUFFER\_FRAME\_FLAG**  
Frame flags on DL\_VID\_BUFFER

### Typedef DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG* **DL\_VID\_BUFFER\_PARAM\_COLOUR\_FLAG**  
Buffer param colour flag

### Typedef DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE* **DL\_VID\_BUFFER\_PARAM\_COLOUR\_RANGE**  
Buffer param colour range

### Typedef DL\_VID\_BUFFER\_PARAM\_TYPE

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_BUFFER\_PARAM\_TYPE* **DL\_VID\_BUFFER\_PARAM\_TYPE**  
Buffer param type

### Typedef DL\_VID\_CHROMA\_FORMAT

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_CHROMA\_FORMAT* **DL\_VID\_CHROMA\_FORMAT**

### Typedef DL\_VID\_CODEC\_FORMAT

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_CODEC\_FORMAT* **DL\_VID\_CODEC\_FORMAT**  
dVID codec format

### Typedef DL\_VID\_DEVICE

- Defined in file dlvid.h

### Typedef Documentation

**typedef void \*****DL\_VID\_DEVICE**  
dVID device handle

### Typedef DL\_VID\_DEVICE\_TYPE

- Defined in file dlvid.h

### Typedef Documentation

**typedef enum** *DL\_VID\_DEVICE\_TYPE* **DL\_VID\_DEVICE\_TYPE**  
dVID device types

---

## Typedef DL\_VID\_RATE\_CONTROL\_MODE

- Defined in file dlvid.h

## Typedef Documentation

**typedef enum** *DL\_VID\_RATE\_CONTROL\_MODE* **DL\_VID\_RATE\_CONTROL\_MODE**  
Rate control mode

## Typedef DL\_VID\_RESULT

- Defined in file dlvid.h

## Typedef Documentation

**typedef enum** *DL\_VID\_RESULT* **DL\_VID\_RESULT**  
dVID result values

## Typedef DL\_VID\_SESSION

- Defined in file dlvid.h

## Typedef Documentation

**typedef void \*****DL\_VID\_SESSION**  
dVID session handle

## Typedef DL\_VID\_SESSION\_STATE

- Defined in file dlvid.h

## Typedef Documentation

**typedef enum** *DL\_VID\_SESSION\_STATE* **DL\_VID\_SESSION\_STATE**  
Session state

- genindex
- search

登临科技保密材料

## INDEX

## D

- DL\_VID\_BUFFER (C++ type), 46
- DL\_VID\_BUFFER\_ALIGNMENT (C macro), 35
- DL\_VID\_BUFFER\_CREATE\_INFO (C++ struct), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::afbcAllocBytes (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::afbcWidthInSuperblocks (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::device (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::flags (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::format (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::height (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::size (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::stride (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::strideAlign (C++ member), 4
- DL\_VID\_BUFFER\_CREATE\_INFO::width (C++ member), 4
- DL\_VID\_BUFFER\_DIRECTION (C++ enum), 11
- DL\_VID\_BUFFER\_DIRECTION (C++ type), 46
- DL\_VID\_BUFFER\_DIRECTION::DL\_VID\_BUFFER\_IN (C++ enumerator), 11
- DL\_VID\_BUFFER\_DIRECTION::DL\_VID\_BUFFER\_OUT (C++ enumerator), 11
- DL\_VID\_BUFFER\_FLAG (C++ enum), 12
- DL\_VID\_BUFFER\_FLAG (C++ type), 46
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_CODECONF (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_DATA\_CORRUPT (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_DECODE\_ONLY (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_ENDOFFRAME (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_EOS (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_EXTRADATA (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_NDOFSUBFRAME (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_READONLY (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_SKIPFRAME (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_STARTTIME (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_SYNCFRAME (C++ enumerator), 12
- DL\_VID\_BUFFER\_FLAG::DL\_VID\_BUFFER\_FLAG\_TIMESTAMPING (C++ enumerator), 12
- DL\_VID\_BUFFER\_FORMAT (C++ enum), 12
- DL\_VID\_BUFFER\_FORMAT (C++ type), 47
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_ABGR\_8888 (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_ARGB\_8888 (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_BGRA\_8888 (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_BITSTREAM (C++ enumerator), 12
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_RGBA\_8888 (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YUV420\_A (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YUV420\_B (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YUV420\_C (C++ enumerator), 12
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YUV420\_P (C++ enumerator), 12
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YUV422\_A (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YUV422\_B (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YUYVYV (C++ enumerator), 13
- DL\_VID\_BUFFER\_FORMAT::DL\_VID\_BUFFER\_FORMAT\_YV12 (C++ enumerator), 13

---

DL_VID_BUFFER_FORMAT::DL_VID_BUFFER_FORMAT_VVD420SEMIPLANAR	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::transferCh	(C++ member), 5
DL_VID_BUFFER_FORMAT::DL_VID_BUFFER_FORMAT_VVD420	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_FLAG	(C++ enum), 14
DL_VID_BUFFER_FRAME_FLAG (C++ enum), 13		DL_VID_BUFFER_PARAM_COLOUR_FLAG	(C++ type), 47
DL_VID_BUFFER_FRAME_FLAG (C++ type), 47		DL_VID_BUFFER_PARAM_COLOUR_FLAG::DL_VID_BUFFER_P	(C++ enumerator), 14
DL_VID_BUFFER_FRAME_FLAG::DL_VID_BUFFER_FRAME_FLAG	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_FLAG::DL_VID_BUFFER_P	(C++ enumerator), 14
DL_VID_BUFFER_FRAME_FLAG::DL_VID_BUFFER_FRAME_FLAG	(C++ enumerator), 14	DL_VID_BUFFER_PARAM_COLOUR_RANGE	(C++ enum), 14
DL_VID_BUFFER_FRAME_FLAG::DL_VID_BUFFER_FRAME_FLAG	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_RANGE	(C++ type), 47
DL_VID_BUFFER_FRAME_FLAG::DL_VID_BUFFER_FRAME_FLAG	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_RANGE::DL_VID_BUFFER_P	(C++ enumerator), 14
DL_VID_BUFFER_FRAME_FLAG::DL_VID_BUFFER_FRAME_FLAG	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_RANGE::DL_VID_BUFFER_P	(C++ enumerator), 14
DL_VID_BUFFER_FRAME_FLAG::DL_VID_BUFFER_FRAME_FLAG	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_RANGE::DL_VID_BUFFER_P	(C++ enumerator), 14
DL_VID_BUFFER_FRAME_FLAG::DL_VID_BUFFER_FRAME_FLAG	(C++ enumerator), 13	DL_VID_BUFFER_PARAM_COLOUR_RANGE::DL_VID_BUFFER_P	(C++ enumerator), 14
DL_VID_BUFFER_PARAM (C++ struct), 5		DL_VID_BUFFER_PARAM_DISPLAY_SIZE	(C++ struct), 6
DL_VID_BUFFER_PARAM::arg (C++ member), 5		DL_VID_BUFFER_PARAM_DISPLAY_SIZE::displayHeight	(C++ member), 6
DL_VID_BUFFER_PARAM::colourDescription	(C++ member), 5	DL_VID_BUFFER_PARAM_DISPLAY_SIZE::displayWidth	(C++ member), 6
DL_VID_BUFFER_PARAM::data (C++ member), 5		DL_VID_BUFFER_PARAM_TYPE (C++ enum), 15	
DL_VID_BUFFER_PARAM::displaySize (C++ member), 5		DL_VID_BUFFER_PARAM_TYPE (C++ type), 48	
DL_VID_BUFFER_PARAM::type (C++ member), 5		DL_VID_BUFFER_PARAM_TYPE::DL_VID_BUFFER_PARAM_TYPE	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION	(C++ struct), 5	DL_VID_BUFFER_PARAM_TYPE::DL_VID_BUFFER_PARAM_TYPE	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::arg (C++ member), 6		DL_VID_BUFFER_PARAM_TYPE::DL_VID_BUFFER_PARAM_TYPE	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_BUFFER_PARAM_TYPE::DL_VID_BUFFER_PARAM_TYPE	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CHROMA_FORMAT (C++ enum), 15	
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CHROMA_FORMAT (C++ type), 48	
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CHROMA_FORMAT_420	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CHROMA_FORMAT_422	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CHROMA_FORMAT_440	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CHROMA_FORMAT_ARGB	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CHROMA_FORMAT_MONO	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 6	DL_VID_CODEC_FORMAT (C++ enum), 15	
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 6	DL_VID_CODEC_FORMAT (C++ type), 48	
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 6	DL_VID_CODEC_FORMAT_DECODE_AV	(C++ enumerator), 16
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 6	DL_VID_CODEC_FORMAT_DECODE_H2	(C++ enumerator), 15
DL_VID_BUFFER_PARAM_COLOUR_DESCRIPTION::DL_VID_BUFFER_P	(C++ member), 5	DL_VID_CODEC_FORMAT_DECODE_H2	(C++ enumerator), 15

---

DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM5 (C++ enumerator), 16	DL_VID_FRAME_ALLOC_PARAMS::planarAllocFrameWidth (C++ member), 7
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM2H264_1 (C++ enumerator), 15	DL_VID_LEVEL_H264_11 (C macro), 35
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM4H264_12 (C++ enumerator), 15	DL_VID_LEVEL_H264_13 (C macro), 35
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM4H264_1b (C++ enumerator), 15	DL_VID_LEVEL_H264_2 (C macro), 36
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM7H264_21 (C++ enumerator), 16	DL_VID_LEVEL_H264_22 (C macro), 36
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM7H264_3 (C++ enumerator), 16	DL_VID_LEVEL_H264_31 (C macro), 37
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM7H264_32 (C++ enumerator), 16	DL_VID_LEVEL_H264_4 (C macro), 37
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM7H265_41 (C++ enumerator), 16	DL_VID_LEVEL_H265_42 (C macro), 37
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM7H265_41 (C++ enumerator), 16	DL_VID_LEVEL_H265_5 (C macro), 38
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM7H265_51 (C++ enumerator), 16	DL_VID_LEVEL_H265_52 (C macro), 38
DL_VID_CODEC_FORMAT::DL_VID_CODEC_FORMAT_DVDRM7H265_6 (C++ enumerator), 16	DL_VID_LEVEL_H265_6 (C macro), 38
DL_VID_DEVICE (C++ type), 48	DL_VID_LEVEL_H265_61 (C macro), 38
DL_VID_DEVICE_INFO (C++ struct), 6	DL_VID_LEVEL_H265_62 (C macro), 39
DL_VID_DEVICE_INFO::channelMask (C++ member), 6	DL_VID_LEVEL_H265_HIGH_TIER_1 (C macro), 39
DL_VID_DEVICE_INFO::clusterMask (C++ member), 6	DL_VID_LEVEL_H265_HIGH_TIER_2 (C macro), 39
DL_VID_DEVICE_INFO::deviceTypeMask (C++ member), 6	DL_VID_LEVEL_H265_HIGH_TIER_21 (C macro), 39
DL_VID_DEVICE_PROP (C++ struct), 7	DL_VID_LEVEL_H265_HIGH_TIER_3 (C macro), 39
DL_VID_DEVICE_PROP::channel (C++ member), 7	DL_VID_LEVEL_H265_HIGH_TIER_31 (C macro), 40
DL_VID_DEVICE_PROP::cluster (C++ member), 7	DL_VID_LEVEL_H265_HIGH_TIER_4 (C macro), 40
DL_VID_DEVICE_PROP::deviceType (C++ member), 7	DL_VID_LEVEL_H265_HIGH_TIER_41 (C macro), 40
DL_VID_DEVICE_PROP::dliDeviceId (C++ member), 7	DL_VID_LEVEL_H265_HIGH_TIER_5 (C macro), 40
DL_VID_DEVICE_PROP::hardwareVersion (C++ member), 7	DL_VID_LEVEL_H265_HIGH_TIER_51 (C macro), 40
DL_VID_DEVICE_PROP::indexInChannel (C++ member), 7	DL_VID_LEVEL_H265_HIGH_TIER_52 (C macro), 41
DL_VID_DEVICE_PROP::maxStreamCount (C++ member), 7	DL_VID_LEVEL_H265_HIGH_TIER_6 (C macro), 41
DL_VID_DEVICE_TYPE (C++ enum), 16	DL_VID_LEVEL_H265_HIGH_TIER_61 (C macro), 41
DL_VID_DEVICE_TYPE (C++ type), 48	DL_VID_LEVEL_H265_HIGH_TIER_62 (C macro), 41
DL_VID_DEVICE_TYPE::DL_VID_DEVICE_DECODER (C++ enumerator), 16	DL_VID_LEVEL_H265_MAIN_TIER_1 (C macro), 41
DL_VID_DEVICE_TYPE::DL_VID_DEVICE_ENCODER (C++ enumerator), 16	DL_VID_LEVEL_H265_MAIN_TIER_2 (C macro), 42
DL_VID_FRAME_ALLOC_PARAMS (C++ struct), 7	DL_VID_LEVEL_H265_MAIN_TIER_21 (C macro), 42
DL_VID_FRAME_ALLOC_PARAMS::afbcAllocByteSize (C++ member), 7	
DL_VID_FRAME_ALLOC_PARAMS::afbcWidthInSuperBlocks (C++ member), 7	
DL_VID_FRAME_ALLOC_PARAMS::planarAllocFrameWidth (C++ member), 7	

DL_VID_LEVEL_H265_MAIN_TIER_3 (C macro), 42	DL_VID_RESULT::DL_VID_RESULT_INVALID_DEVICE (C++ enumerator), 17
DL_VID_LEVEL_H265_MAIN_TIER_31 (C macro), 42	DL_VID_RESULT::DL_VID_RESULT_IO_ERROR (C++ enumerator), 18
DL_VID_LEVEL_H265_MAIN_TIER_4 (C macro), 42	DL_VID_RESULT::DL_VID_RESULT_NOT_IMPLEMENTED (C++ enumerator), 17
DL_VID_LEVEL_H265_MAIN_TIER_41 (C macro), 43	DL_VID_RESULT::DL_VID_RESULT_NOT_READY (C++ enumerator), 17
DL_VID_LEVEL_H265_MAIN_TIER_5 (C macro), 43	DL_VID_RESULT::DL_VID_RESULT_SESSION_HANG (C++ enumerator), 18
DL_VID_LEVEL_H265_MAIN_TIER_51 (C macro), 43	DL_VID_RESULT::DL_VID_RESULT_SESSION_LIMIT (C++ enumerator), 18
DL_VID_LEVEL_H265_MAIN_TIER_52 (C macro), 43	DL_VID_RESULT::DL_VID_RESULT_STREAM_CORRUPT (C++ enumerator), 17
DL_VID_LEVEL_H265_MAIN_TIER_6 (C macro), 43	DL_VID_RESULT::DL_VID_RESULT_STREAM_NOT_SUPPORTED (C++ enumerator), 18
DL_VID_LEVEL_H265_MAIN_TIER_61 (C macro), 44	DL_VID_RESULT::DL_VID_RESULT_SUCCESS (C++ enumerator), 17
DL_VID_LEVEL_H265_MAIN_TIER_62 (C macro), 44	DL_VID_RESULT::DL_VID_RESULT_TIMEOUT (C++ enumerator), 17
DL_VID_PROFILE_H264_BASELINE (C macro), 44	DL_VID_RESULT::DL_VID_RESULT_UNDEFINED_ERROR (C++ enumerator), 17
DL_VID_PROFILE_H264_HIGH (C macro), 44	DL_VID_RESULT::DL_VID_RESULT_VERSION_MISMATCH (C++ enumerator), 17
DL_VID_PROFILE_H264_HIGH_10 (C macro), 44	DL_VID_ROI_COUNT_MAX (C macro), 46
DL_VID_PROFILE_H264_MAIN (C macro), 45	DL_VID_ROI_REGION (C++ struct), 8
DL_VID_PROFILE_H265_MAIN (C macro), 45	DL_VID_ROI_REGION::mbxLeft (C++ member), 8
DL_VID_PROFILE_H265_MAIN_10 (C macro), 45	DL_VID_ROI_REGION::mbxRight (C++ member), 8
DL_VID_PROFILE_H265_MAIN_INTRA (C macro), 45	DL_VID_ROI_REGION::mbyBottom (C++ member), 8
DL_VID_PROFILE_H265_MAIN_STILL (C macro), 45	DL_VID_ROI_REGION::mbyTop (C++ member), 8
DL_VID_RATE_CONTROL_MODE (C++ enum), 16	DL_VID_SEQUENCE_PARAMS (C++ struct), 8
DL_VID_RATE_CONTROL_MODE (C++ type), 49	DL_VID_SEQUENCE_PARAMS::bitdepthChroma (C++ member), 8
DL_VID_RATE_CONTROL_MODE::DL_VID_RATE_CONTROL_MODE_CONSTANT (C++ enumerator), 17	DL_VID_SEQUENCE_PARAMS::bitdepthLuma (C++ member), 8
DL_VID_RATE_CONTROL_MODE::DL_VID_RATE_CONTROL_MODE_LINEAR (C++ enumerator), 16	DL_VID_SEQUENCE_PARAMS::chromaFormat (C++ member), 8
DL_VID_RATE_CONTROL_MODE::DL_VID_RATE_CONTROL_MODE_NON_LINEAR (C++ enumerator), 16	DL_VID_SEQUENCE_PARAMS::interlace (C++ member), 8
DL_VID_RATE_CONTROL_MODE::DL_VID_RATE_CONTROL_MODE_VARIABLE (C++ enumerator), 17	DL_VID_SEQUENCE_PARAMS::numAfbcBuffers (C++ member), 8
DL_VID_RESULT (C++ enum), 17	DL_VID_SESSION (C++ type), 49
DL_VID_RESULT (C++ type), 49	DL_VID_SESSION_CREATE_INFO (C++ struct), 9
DL_VID_RESULT::DL_VID_RESULT_BAD_BUFFER_DIRECTION (C++ enumerator), 17	DL_VID_SESSION_CREATE_INFO::codecFormat (C++ member), 9
DL_VID_RESULT::DL_VID_RESULT_BAD_PARAMETER (C++ enumerator), 17	DL_VID_SESSION_CREATE_INFO::device (C++ member), 9
DL_VID_RESULT::DL_VID_RESULT_FAILED (C++ enumerator), 17	DL_VID_SESSION_STATE (C++ enum), 18
DL_VID_RESULT::DL_VID_RESULT_HARDWARE_ERROR (C++ enumerator), 17	DL_VID_SESSION_STATE (C++ type), 49
DL_VID_RESULT::DL_VID_RESULT_HARDWARE_INTERRUPT_ERROR (C++ enumerator), 17	
DL_VID_RESULT::DL_VID_RESULT_INSUFFICIENT_RESOURCES (C++ enumerator), 17	



DL\_VID\_SESSION\_STATE::DL\_VID\_SESSION\_STATE\_PENDING  
(C++ *enumerator*), 18  
DL\_VID\_SESSION\_STATE::DL\_VID\_SESSION\_STATE\_RUNNING  
(C++ *enumerator*), 18  
DL\_VID\_SESSION\_STATE::DL\_VID\_SESSION\_STATE\_STOPPED  
(C++ *enumerator*), 18  
DLVIDAPI (C *macro*), 46