Memory-to-Memory Stateful Video Encoder Interface

A stateful video encoder takes raw video frames in display order and encodes them into a bytestream. It generates complete chunks of the bytestream, including all metadata, headers, etc. The resulting bytestream does not require any further post-processing by the client.

Performing software stream processing, header generation etc. in the driver in order to support this interface is strongly discouraged. In case such operations are needed, use of the Stateless Video Encoder Interface (in development) is strongly advised.

Conventions and Notations Used in This Document

1. The general V4L2 API rules apply if not specified in this document otherwise.
2. The meaning of words "must", "may", "should", etc. is as per RFC 2119.
3. All steps not marked "optional" are required.
4. :c:func:`VIDIOC_G_EXT_CTRLS` and :c:func:`VIDIOC_S_EXT_CTRLS` may be used interchangeably with :c:func:`VIDIOC_G_CTRL` and :c:func:`VIDIOC_S_CTRL`, unless specified otherwise.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 30); *backlink*

   Unknown interpreted text role "c:func".

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 30); *backlink*

   Unknown interpreted text role "c:func".

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 30); *backlink*

   Unknown interpreted text role "c:func".

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 30); *backlink*

   Unknown interpreted text role "c:func".

5. Single-planar API (see :ref:`planar-apis`) and applicable structures may be used interchangeably with multi-planar API, unless specified otherwise, depending on encoder capabilities and following the general V4L2 guidelines.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 34); *backlink*

   Unknown interpreted text role "ref".

6. i = [a..b]: sequence of integers from a to b, inclusive, i.e. i = [0..2]: i = 0, 1, 2.
7. Given an `OUTPUT` buffer A, then A' represents a buffer on the `CAPTURE` queue containing data that resulted from processing buffer A.

Glossary

Refer to :ref:`decoder-glossary`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 47); *backlink*

Unknown interpreted text role "ref".

State Machine

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 52)

Unknown directive type "kernel-render".

```
.. kernel-render:: DOT
   :alt: DOT digraph of encoder state machine
   :caption: Encoder State Machine

   digraph encoder_state_machine {
       node [shape = doublecircle, label="Encoding"] Encoding;

       node [shape = circle, label="Initialization"] Initialization;
       node [shape = circle, label="Stopped"] Stopped;
       node [shape = circle, label="Drain"] Drain;
       node [shape = circle, label="Reset"] Reset;

       node [shape = point]; qi
       qi -> Initialization [ label = "open()" ];

       Initialization -> Encoding [ label = "Both queues streaming" ];

       Encoding -> Drain [ label = "V4L2_ENC_CMD_STOP" ];
       Encoding -> Reset [ label = "VIDIOC_STREAMOFF(CAPTURE)" ];
       Encoding -> Stopped [ label = "VIDIOC_STREAMOFF(OUTPUT)" ];
       Encoding -> Encoding;

       Drain -> Stopped [ label = "All CAPTURE\nbuffers dequeued\nor\nVIDIOC_STREAMOFF(OUTPUT)" ];
       Drain -> Reset [ label = "VIDIOC_STREAMOFF(CAPTURE)" ];

       Reset -> Encoding [ label = "VIDIOC_STREAMON(CAPTURE)" ];
       Reset -> Initialization [ label = "VIDIOC_REQBUFS(OUTPUT, 0)" ];

       Stopped -> Encoding [ label = "V4L2_ENC_CMD_START\nor\nVIDIOC_STREAMON(OUTPUT)" ];
       Stopped -> Reset [ label = "VIDIOC_STREAMOFF(CAPTURE)" ];
   }
```

Querying Capabilities

1. To enumerate the set of coded formats supported by the encoder, the client may call :c:func:`VIDIOC_ENUM_FMT` on `CAPTURE`.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 87); *backlink*

   Unknown interpreted text role "c:func".

   - The full set of supported formats will be returned, regardless of the format set on `OUTPUT`.

2. To enumerate the set of supported raw formats, the client may call :c:func:`VIDIOC_ENUM_FMT` on `OUTPUT`.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 93); *backlink*

   Unknown interpreted text role "c:func".

   - Only the formats supported for the format currently active on `CAPTURE` will be returned.
   - In order to enumerate raw formats supported by a given coded format, the client must first set that coded format on `CAPTURE` and then enumerate the formats on `OUTPUT`.

3. The client may use :c:func:`VIDIOC_ENUM_FRAMESIZES` to detect supported resolutions for a given format, passing the desired pixel format in :c:type:`v4l2_frmsizeenum` `pixel_format`.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 103); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 103**); ***backlink***

Unknown interpreted text role "c:type".

- Values returned by :c:func:`VIDIOC_ENUM_FRAMESIZES` for a coded pixel format will include all possible coded resolutions supported by the encoder for the given coded pixel format.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 107**); ***backlink***

  Unknown interpreted text role "c:func".

- Values returned by :c:func:`VIDIOC_ENUM_FRAMESIZES` for a raw pixel format will include all possible frame buffer resolutions supported by the encoder for the given raw pixel format and coded format currently set on `CAPTURE`.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 111**); ***backlink***

  Unknown interpreted text role "c:func".

4. The client may use :c:func:`VIDIOC_ENUM_FRAMEINTERVALS` to detect supported frame intervals for a given format and resolution, passing the desired pixel format in :c:type:`v4l2_frmsizeenum` `pixel_format` and the resolution in :c:type:`v4l2_frmsizeenum` `width` and :c:type:`v4l2_frmsizeenum` `height`.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 116**); ***backlink***

   Unknown interpreted text role "c:func".

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 116**); ***backlink***

   Unknown interpreted text role "c:type".

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 116**); ***backlink***

   Unknown interpreted text role "c:type".

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 116**); ***backlink***

   Unknown interpreted text role "c:type".

   - Values returned by :c:func:`VIDIOC_ENUM_FRAMEINTERVALS` for a coded pixel format and coded resolution will include all possible frame intervals supported by the encoder for the given coded pixel format and resolution.

     System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 122**); ***backlink***

     Unknown interpreted text role "c:func".

   - Values returned by :c:func:`VIDIOC_ENUM_FRAMEINTERVALS` for a raw pixel format and resolution will include all possible frame intervals supported by the encoder for the given raw pixel format and resolution and for the coded format, coded resolution and coded frame interval currently set on `CAPTURE`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 126**); ***backlink***
>
> Unknown interpreted text role "c:func".

- Support for :c:func:`VIDIOC_ENUM_FRAMEINTERVALS` is optional. If it is not implemented, then there are no special restrictions other than the limits of the codec itself.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 132**); ***backlink***
>
> Unknown interpreted text role "c:func".

5. Supported profiles and levels for the coded format currently set on `CAPTURE`, if applicable, may be queried using their respective controls via :c:func:`VIDIOC_QUERYCTRL`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 136**); ***backlink***
>
> Unknown interpreted text role "c:func".

6. Any additional encoder capabilities may be discovered by querying their respective controls.

Initialization

1. Set the coded format on the `CAPTURE` queue via :c:func:`VIDIOC_S_FMT`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, **line 146**); ***backlink***
>
> Unknown interpreted text role "c:func".

- **Required fields:**

  `type`
  : a `V4L2_BUF_TYPE_*` enum appropriate for `CAPTURE`.

  `pixelformat`
  : the coded format to be produced.

  `sizeimage`
  : desired size of `CAPTURE` buffers; the encoder may adjust it to match hardware requirements.

  `width`, `height`
  : ignored (read-only).

  other fields
  : follow standard semantics.

- **Return fields:**

  `sizeimage`
  : adjusted size of `CAPTURE` buffers.

  `width`, `height`
  : the coded size selected by the encoder based on current state, e.g. `OUTPUT` format, selection rectangles, etc. (read-only).

> **Important**
>
> Changing the `CAPTURE` format may change the currently set `OUTPUT` format. How the new `OUTPUT` format is determined is up to the encoder and the client must ensure it matches its needs afterwards.

2. **Optional.** Enumerate supported `OUTPUT` formats (raw formats for source) for the selected coded format via :c:func:`VIDIOC_ENUM_FMT`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-

resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 181); *backlink*

Unknown interpreted text role "c:func".

- **Required fields:**

  `type`
  : a `V4L2_BUF_TYPE_*` enum appropriate for `OUTPUT`.

  other fields
  : follow standard semantics.

- **Return fields:**

  `pixelformat`
  : raw format supported for the coded format currently selected on the `CAPTURE` queue.

  other fields
  : follow standard semantics.

3. Set the raw source format on the `OUTPUT` queue via :c:func:`VIDIOC_S_FMT`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 201); *backlink*

Unknown interpreted text role "c:func".

- **Required fields:**

  `type`
  : a `V4L2_BUF_TYPE_*` enum appropriate for `OUTPUT`.

  `pixelformat`
  : raw format of the source.

  `width`, `height`
  : source resolution.

  other fields
  : follow standard semantics.

- **Return fields:**

  `width`, `height`
  : may be adjusted to match encoder minimums, maximums and alignment requirements, as required by the currently selected formats, as reported by :c:func:`VIDIOC_ENUM_FRAMESIZES`.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 221); *backlink*

  Unknown interpreted text role "c:func".

  other fields
  : follow standard semantics.

- Setting the `OUTPUT` format will reset the selection rectangles to their default values, based on the new resolution, as described in the next step.

4. Set the raw frame interval on the `OUTPUT` queue via :c:func:`VIDIOC_S_PARM`. This also sets the coded frame interval on the `CAPTURE` queue to the same value.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 232); *backlink*

Unknown interpreted text role "c:func".

- ** Required fields:**

  `type`
  : a `V4L2_BUF_TYPE_*` enum appropriate for `OUTPUT`.

  `parm.output`
  : set all fields except `parm.output.timeperframe` to 0.

`parm.output.timeperframe`
: the desired frame interval; the encoder may adjust it to match hardware requirements.

- **Return fields:**

`parm.output.timeperframe`
: the adjusted frame interval.

> **Important**
>
> Changing the `OUTPUT` frame interval *also* sets the framerate that the encoder uses to encode the video. So setting the frame interval to 1/24 (or 24 frames per second) will produce a coded video stream that can be played back at that speed. The frame interval for the `OUTPUT` queue is just a hint, the application may provide raw frames at a different rate. It can be used by the driver to help schedule multiple encoders running in parallel.
>
> In the next step the `CAPTURE` frame interval can optionally be changed to a different value. This is useful for off-line encoding were the coded frame interval can be different from the rate at which raw frames are supplied.

> **Important**
>
> `timeperframe` deals with *frames*, not fields. So for interlaced formats this is the time per two fields, since a frame consists of a top and a bottom field.

> **Note**
>
> It is due to historical reasons that changing the `OUTPUT` frame interval also changes the coded frame interval on the `CAPTURE` queue. Ideally these would be independent settings, but that would break the existing API.

5. **Optional** Set the coded frame interval on the `CAPTURE` queue via :c:func:`VIDIOC_S_PARM`. This is only necessary if the coded frame interval is different from the raw frame interval, which is typically the case for off-line encoding. Support for this feature is signalled by the :ref:`V4L2_FMT_FLAG_ENC_CAP_FRAME_INTERVAL <fmtdesc-flags>` format flag.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 281); *backlink*
>
> Unknown interpreted text role "c:func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 281); *backlink*
>
> Unknown interpreted text role "ref".

- ** Required fields:**

`type`
: a `V4L2_BUF_TYPE_*` enum appropriate for `CAPTURE`.

`parm.capture`
: set all fields except `parm.capture.timeperframe` to 0.

`parm.capture.timeperframe`
: the desired coded frame interval; the encoder may adjust it to match hardware requirements.

- **Return fields:**

`parm.capture.timeperframe`
: the adjusted frame interval.

> **Important**
>
> Changing the `CAPTURE` frame interval sets the framerate for the coded video. It does *not* set the rate at which buffers arrive on the `CAPTURE` queue, that depends on how fast the encoder is and how fast raw frames are queued on the `OUTPUT` queue.

> **Important**
>
> `timeperframe` deals with *frames*, not fields. So for interlaced formats this is the time per two fields, since a frame consists of a top and a bottom field.

Note

Not all drivers support this functionality, in that case just set the desired coded frame interval for the `OUTPUT` queue.

However, drivers that can schedule multiple encoders based on the `OUTPUT` frame interval must support this optional feature.

6. **Optional.** Set the visible resolution for the stream metadata via :c:func:`VIDIOC_S_SELECTION` on the `OUTPUT` queue if it is desired to be different than the full OUTPUT resolution.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 325); ***backlink***

   Unknown interpreted text role "c:func".

   - **Required fields:**

     `type`
     : a `V4L2_BUF_TYPE_*` enum appropriate for `OUTPUT`.

     `target`
     : set to `V4L2_SEL_TGT_CROP`.

     `r.left`, `r.top`, `r.width`, `r.height`
     : visible rectangle; this must fit within the *V4L2_SEL_TGT_CROP_BOUNDS* rectangle and may be subject to adjustment to match codec and hardware constraints.

   - **Return fields:**

     `r.left`, `r.top`, `r.width`, `r.height`
     : visible rectangle adjusted by the encoder.

   - The following selection targets are supported on `OUTPUT`:

     `V4L2_SEL_TGT_CROP_BOUNDS`
     : equal to the full source frame, matching the active `OUTPUT` format.

     `V4L2_SEL_TGT_CROP_DEFAULT`
     : equal to `V4L2_SEL_TGT_CROP_BOUNDS`.

     `V4L2_SEL_TGT_CROP`
     : rectangle within the source buffer to be encoded into the `CAPTURE` stream; defaults to `V4L2_SEL_TGT_CROP_DEFAULT`.

       Note

       A common use case for this selection target is encoding a source video with a resolution that is not a multiple of a macroblock, e.g. the common 1920x1080 resolution may require the source buffers to be aligned to 1920x1088 for codecs with 16x16 macroblock size. To avoid encoding the padding, the client needs to explicitly configure this selection target to 1920x1080.

   Warning

   The encoder may adjust the crop/compose rectangles to the nearest supported ones to meet codec and hardware requirements. The client needs to check the adjusted rectangle returned by :c:func:`VIDIOC_S_SELECTION`.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 371); ***backlink***

   Unknown interpreted text role "c:func".

7. Allocate buffers for both `OUTPUT` and `CAPTURE` via :c:func:`VIDIOC_REQBUFS`. This may be performed in any order.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, **line 375**); ***backlink***

Unknown interpreted text role "c:func".

- **Required fields:**

  `count`
  : requested number of buffers to allocate; greater than zero.

  `type`
  : a `V4L2_BUF_TYPE_*` enum appropriate for `OUTPUT` or `CAPTURE`.

  other fields
  : follow standard semantics.

- **Return fields:**

  `count`
  : actual number of buffers allocated.

Warning

The actual number of allocated buffers may differ from the `count` given. The client must check the updated value of `count` after the call returns.

Note

To allocate more than the minimum number of OUTPUT buffers (for pipeline depth), the client may query the `V4L2_CID_MIN_BUFFERS_FOR_OUTPUT` control to get the minimum number of buffers required, and pass the obtained value plus the number of additional buffers needed in the `count` field to :c:func:`VIDIOC_REQBUFS`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, **line 403**); ***backlink***

Unknown interpreted text role "c:func".

Alternatively, :c:func:`VIDIOC_CREATE_BUFS` can be used to have more control over buffer allocation.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, **line 409**); ***backlink***

Unknown interpreted text role "c:func".

- **Required fields:**

  `count`
  : requested number of buffers to allocate; greater than zero.

  `type`
  : a `V4L2_BUF_TYPE_*` enum appropriate for `OUTPUT`.

  other fields
  : follow standard semantics.

- **Return fields:**

  `count`
  : adjusted to the number of allocated buffers.

8. Begin streaming on both `OUTPUT` and `CAPTURE` queues via :c:func:`VIDIOC_STREAMON`. This may be performed in any order. The actual encoding process starts when both queues start streaming.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, **line 428**); ***backlink***

Unknown interpreted text role "c:func".

Note

If the client stops the `CAPTURE` queue during the encode process and then restarts it again, the encoder will begin generating a stream independent from the stream generated before the stop. The exact constraints depend on the coded format, but may include the following implications:

- encoded frames produced after the restart must not reference any frames produced before the stop, e.g. no long term references for H.264/HEVC,
- any headers that must be included in a standalone stream must be produced again, e.g. SPS and PPS for H.264/HEVC.

Encoding

This state is reached after the *Initialization* sequence finishes successfully. In this state, the client queues and dequeues buffers to both queues via :c:func:`VIDIOC_QBUF` and :c:func:`VIDIOC_DQBUF`, following the standard semantics.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 449); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 449); *backlink*

Unknown interpreted text role "c:func".

The content of encoded `CAPTURE` buffers depends on the active coded pixel format and may be affected by codec-specific extended controls, as stated in the documentation of each format.

Both queues operate independently, following standard behavior of V4L2 buffer queues and memory-to-memory devices. In addition, the order of encoded frames dequeued from the `CAPTURE` queue may differ from the order of queuing raw frames to the `OUTPUT` queue, due to properties of the selected coded format, e.g. frame reordering.

The client must not assume any direct relationship between `CAPTURE` and `OUTPUT` buffers and any specific timing of buffers becoming available to dequeue. Specifically:

- a buffer queued to `OUTPUT` may result in more than one buffer produced on `CAPTURE` (for example, if returning an encoded frame allowed the encoder to return a frame that preceded it in display, but succeeded it in the decode order; however, there may be other reasons for this as well),
- a buffer queued to `OUTPUT` may result in a buffer being produced on `CAPTURE` later into encode process, and/or after processing further `OUTPUT` buffers, or be returned out of order, e.g. if display reordering is used,
- buffers may become available on the `CAPTURE` queue without additional buffers queued to `OUTPUT` (e.g. during drain or `EOS`), because of the `OUTPUT` buffers queued in the past whose encoding results are only available at later time, due to specifics of the encoding process,
- buffers queued to `OUTPUT` may not become available to dequeue instantly after being encoded into a corresponding `CAPTURE` buffer, e.g. if the encoder needs to use the frame as a reference for encoding further frames.

Note

To allow matching encoded `CAPTURE` buffers with `OUTPUT` buffers they originated from, the client can set the `timestamp` field of the :c:type:`v4l2_buffer` struct when queuing an `OUTPUT` buffer. The `CAPTURE` buffer(s), which resulted from encoding that `OUTPUT` buffer will have their `timestamp` field set to the same value when dequeued.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 489); *backlink*

Unknown interpreted text role "c:type".

In addition to the straightforward case of one `OUTPUT` buffer producing one `CAPTURE` buffer, the following cases are defined:

- one `OUTPUT` buffer generates multiple `CAPTURE` buffers: the same `OUTPUT` timestamp will be copied to multiple `CAPTURE` buffers,
- the encoding order differs from the presentation order (i.e. the `CAPTURE` buffers are out-of-order compared to the `OUTPUT` buffers): `CAPTURE` timestamps will not retain the order of `OUTPUT` timestamps.

Note

To let the client distinguish between frame types (keyframes, intermediate frames; the exact list of types depends on the coded format), the `CAPTURE` buffers will have corresponding flag bits set in their :c:type:`v4l2_buffer` struct when dequeued. See the documentation of :c:type:`v4l2_buffer` and each coded pixel format for exact list of flags and their meanings.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 507); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 507); *backlink*

Unknown interpreted text role "c:type".

Should an encoding error occur, it will be reported to the client with the level of details depending on the encoder capabilities. Specifically:

- the `CAPTURE` buffer (if any) that contains the results of the failed encode operation will be returned with the `V4L2_BUF_FLAG_ERROR` flag set,
- if the encoder is able to precisely report the `OUTPUT` buffer(s) that triggered the error, such buffer(s) will be returned with the `V4L2_BUF_FLAG_ERROR` flag set.

Note

If a `CAPTURE` buffer is too small then it is just returned with the `V4L2_BUF_FLAG_ERROR` flag set. More work is needed to detect that this error occurred because the buffer was too small, and to provide support to free existing buffers that were too small.

In case of a fatal failure that does not allow the encoding to continue, any further operations on corresponding encoder file handle will return the -EIO error code. The client may close the file handle and open a new one, or alternatively reinitialize the instance by stopping streaming on both queues, releasing all buffers and performing the Initialization sequence again.

Encoding Parameter Changes

The client is allowed to use :c:func:`VIDIOC_S_CTRL` to change encoder parameters at any time. The availability of parameters is encoder-specific and the client must query the encoder to find the set of available controls.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 540); *backlink*

Unknown interpreted text role "c:func".

The ability to change each parameter during encoding is encoder-specific, as per the standard semantics of the V4L2 control interface. The client may attempt to set a control during encoding and if the operation fails with the -EBUSY error code, the `CAPTURE` queue needs to be stopped for the configuration change to be allowed. To do this, it may follow the *Drain* sequence to avoid losing the already queued/encoded frames.

The timing of parameter updates is encoder-specific, as per the standard semantics of the V4L2 control interface. If the client needs to apply the parameters exactly at specific frame, using the Request API (:ref:`media-request-api`) should be considered, if supported by the encoder.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 551); *backlink*

Unknown interpreted text role "ref".

Drain

To ensure that all the queued `OUTPUT` buffers have been processed and the related `CAPTURE` buffers are given to the client, the client must follow the drain sequence described below. After the drain sequence ends, the client has received all encoded frames for all `OUTPUT` buffers queued before the sequence was started.

1. Begin the drain sequence by issuing :c:func:`VIDIOC_ENCODER_CMD`.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 565); *backlink*

   Unknown interpreted text role "c:func".

   - **Required fields:**

     `cmd`
     : set to `V4L2_ENC_CMD_STOP`.

     `flags`
     : set to 0.

     `pts`
     : set to 0.

   Warning

   The sequence can be only initiated if both `OUTPUT` and `CAPTURE` queues are streaming. For compatibility reasons, the call to :c:func:`VIDIOC_ENCODER_CMD` will not fail even if any of the queues is not streaming, but at the same time it will not initiate the *Drain* sequence and so the steps described below would not be applicable.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 580); *backlink*

   Unknown interpreted text role "c:func".

2. Any `OUTPUT` buffers queued by the client before the :c:func:`VIDIOC_ENCODER_CMD` was issued will be processed and encoded as normal. The client must continue to handle both queues independently, similarly to normal encode operation. This includes:

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 586); *backlink*

   Unknown interpreted text role "c:func".

   - queuing and dequeuing `CAPTURE` buffers, until a buffer marked with the `V4L2_BUF_FLAG_LAST` flag is dequeued,

     Warning

     The last buffer may be empty (with :c:type:`v4l2_buffer` `bytesused` = 0) and in that case it must be ignored by the client, as it does not contain an encoded frame.

     System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master) (Documentation) (userspace-api) (media) (v4l)dev-encoder.rst`, line 596); *backlink*

     Unknown interpreted text role "c:type".

     Note

     Any attempt to dequeue more `CAPTURE` buffers beyond the buffer marked with `V4L2_BUF_FLAG_LAST` will result in a -EPIPE error from :c:func:`VIDIOC_DQBUF`.

     System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 602); *backlink*

Unknown interpreted text role "c:func".

- dequeuing processed `OUTPUT` buffers, until all the buffers queued before the `V4L2_ENC_CMD_STOP` command are dequeued,
- dequeuing the `V4L2_EVENT_EOS` event, if the client subscribes to it.

Note

For backwards compatibility, the encoder will signal a `V4L2_EVENT_EOS` event when the last frame has been encoded and all frames are ready to be dequeued. It is deprecated behavior and the client must not rely on it. The `V4L2_BUF_FLAG_LAST` buffer flag should be used instead.

3. Once all `OUTPUT` buffers queued before the `V4L2_ENC_CMD_STOP` call are dequeued and the last `CAPTURE` buffer is dequeued, the encoder is stopped and it will accept, but not process any newly queued `OUTPUT` buffers until the client issues any of the following operations:
   - `V4L2_ENC_CMD_START` - the encoder will not be reset and will resume operation normally, with all the state from before the drain,
   - a pair of :c:func:`VIDIOC_STREAMOFF` and :c:func:`VIDIOC_STREAMON` on the `CAPTURE` queue - the encoder will be reset (see the *Reset* sequence) and then resume encoding,

     System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 626); *backlink*

     Unknown interpreted text role "c:func".

     System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 626); *backlink*

     Unknown interpreted text role "c:func".

   - a pair of :c:func:`VIDIOC_STREAMOFF` and :c:func:`VIDIOC_STREAMON` on the `OUTPUT` queue - the encoder will resume operation normally, however any source frames queued to the `OUTPUT` queue between `V4L2_ENC_CMD_STOP` and :c:func:`VIDIOC_STREAMOFF` will be discarded.

     System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 630); *backlink*

     Unknown interpreted text role "c:func".

     System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 630); *backlink*

     Unknown interpreted text role "c:func".

     System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst, line 630); *backlink*

     Unknown interpreted text role "c:func".

Note

Once the drain sequence is initiated, the client needs to drive it to completion, as described by the steps above, unless it aborts the process by issuing :c:func:`VIDIOC_STREAMOFF` on any of the `OUTPUT` or `CAPTURE` queues. The client is not allowed to issue `V4L2_ENC_CMD_START` or `V4L2_ENC_CMD_STOP` again while the drain sequence is in progress and they will fail with -EBUSY error code if attempted.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 637); *backlink*

Unknown interpreted text role "c:func".

For reference, handling of various corner cases is described below:

- In case of no buffer in the `OUTPUT` queue at the time the `V4L2_ENC_CMD_STOP` command was issued, the drain sequence completes immediately and the encoder returns an empty `CAPTURE` buffer with the `V4L2_BUF_FLAG_LAST` flag set.
- In case of no buffer in the `CAPTURE` queue at the time the drain sequence completes, the next time the client queues a `CAPTURE` buffer it is returned at once as an empty buffer with the `V4L2_BUF_FLAG_LAST` flag set.
- If :c:func:`VIDIOC_STREAMOFF` is called on the `CAPTURE` queue in the middle of the drain sequence, the drain sequence is canceled and all `CAPTURE` buffers are implicitly returned to the client.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 656); *backlink*

  Unknown interpreted text role "c:func".

- If :c:func:`VIDIOC_STREAMOFF` is called on the `OUTPUT` queue in the middle of the drain sequence, the drain sequence completes immediately and next `CAPTURE` buffer will be returned empty with the `V4L2_BUF_FLAG_LAST` flag set.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 660); *backlink*

  Unknown interpreted text role "c:func".

Although not mandatory, the availability of encoder commands may be queried using :c:func:`VIDIOC_TRY_ENCODER_CMD`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 665); *backlink*

Unknown interpreted text role "c:func".

Reset

The client may want to request the encoder to reinitialize the encoding, so that the following stream data becomes independent from the stream data generated before. Depending on the coded format, that may imply that:

- encoded frames produced after the restart must not reference any frames produced before the stop, e.g. no long term references for H.264/HEVC,
- any headers that must be included in a standalone stream must be produced again, e.g. SPS and PPS for H.264/HEVC.

This can be achieved by performing the reset sequence.

1. Perform the *Drain* sequence to ensure all the in-flight encoding finishes and respective buffers are dequeued.
2. Stop streaming on the `CAPTURE` queue via :c:func:`VIDIOC_STREAMOFF`. This will return all currently queued `CAPTURE` buffers to the client, without valid frame data.

   System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 686); *backlink*

   Unknown interpreted text role "c:func".

3. Start streaming on the `CAPTURE` queue via :c:func:`VIDIOC_STREAMON` and continue with regular encoding sequence. The encoded frames produced into `CAPTURE` buffers from now on will contain a standalone stream that can be decoded

without the need for frames encoded before the reset sequence, starting at the first `OUTPUT` buffer queued after issuing the *V4L2_ENC_CMD_STOP* of the *Drain* sequence.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)dev-encoder.rst`, line 690); *backlink*

Unknown interpreted text role "c:func".

This sequence may be also used to change encoding parameters for encoders without the ability to change the parameters on the fly.

Commit Points

Setting formats and allocating buffers triggers changes in the behavior of the encoder.

1. Setting the format on the `CAPTURE` queue may change the set of formats supported/advertised on the `OUTPUT` queue. In particular, it also means that the `OUTPUT` format may be reset and the client must not rely on the previously set format being preserved.
2. Enumerating formats on the `OUTPUT` queue always returns only formats supported for the current `CAPTURE` format.
3. Setting the format on the `OUTPUT` queue does not change the list of formats available on the `CAPTURE` queue. An attempt to set the `OUTPUT` format that is not supported for the currently selected `CAPTURE` format will result in the encoder adjusting the requested `OUTPUT` format to a supported one.
4. Enumerating formats on the `CAPTURE` queue always returns the full set of supported coded formats, irrespective of the current `OUTPUT` format.
5. While buffers are allocated on any of the `OUTPUT` or `CAPTURE` queues, the client must not change the format on the `CAPTURE` queue. Drivers will return the -EBUSY error code for any such format change attempt.

To summarize, setting formats and allocation must always start with the `CAPTURE` queue and the `CAPTURE` queue is the master that governs the set of supported formats for the `OUTPUT` queue.