Memory-to-Memory Stateful Video Decoder Interface

A stateful video decoder takes complete chunks of the bytestream (e.g. Annex-B H.264/HEVC stream, raw VP8/9 stream) and decodes them into raw video frames in display order. The decoder is expected not to require any additional information from the client to process these buffers.

Performing software parsing, processing etc. of the stream in the driver in order to support this interface is strongly discouraged. In case such operations are needed, use of the Stateless Video Decoder 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-decoder.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-decoder.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-decoder.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-decoder.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 decoder 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-decoder.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

CAPTURE

the destination buffer queue; for decoders, the queue of buffers containing decoded frames; for encoders, the queue of buffers containing an encoded bytestream; `V4L2_BUF_TYPE_VIDEO_CAPTURE` or `V4L2_BUF_TYPE_VIDEO_CAPTURE_MPLANE`; data is captured from the hardware into `CAPTURE` buffers.

client

the application communicating with the decoder or encoder implementing this interface.

coded format

encoded/compressed video bytestream format (e.g. H.264, VP8, etc.); see also: raw format.

coded height

height for given coded resolution.

coded resolution

stream resolution in pixels aligned to codec and hardware requirements; typically visible resolution rounded up to full macroblocks; see also: visible resolution.

coded width

width for given coded resolution.

decode order

the order in which frames are decoded; may differ from display order if the coded format includes a feature of frame reordering; for decoders, `OUTPUT` buffers must be queued by the client in decode order; for encoders `CAPTURE` buffers must be returned by the encoder in decode order.

destination

data resulting from the decode process; see `CAPTURE`.

display order

the order in which frames must be displayed; for encoders, `OUTPUT` buffers must be queued by the client in display order; for decoders, `CAPTURE` buffers must be returned by the decoder in display order.

DPB

Decoded Picture Buffer; an H.264/HEVC term for a buffer that stores a decoded raw frame available for reference in further decoding steps.

EOS

end of stream.

IDR

Instantaneous Decoder Refresh; a type of a keyframe in an H.264/HEVC-encoded stream, which clears the list of earlier reference frames (DPBs).

keyframe

an encoded frame that does not reference frames decoded earlier, i.e. can be decoded fully on its own.

macroblock

a processing unit in image and video compression formats based on linear block transforms (e.g. H.264, VP8, VP9); codec-specific, but for most of popular codecs the size is 16x16 samples (pixels).

OUTPUT

the source buffer queue; for decoders, the queue of buffers containing an encoded bytestream; for encoders, the queue of buffers containing raw frames; `V4L2_BUF_TYPE_VIDEO_OUTPUT` or `V4L2_BUF_TYPE_VIDEO_OUTPUT_MPLANE`; the hardware is fed with data from `OUTPUT` buffers.

PPS

Picture Parameter Set; a type of metadata entity in an H.264/HEVC bytestream.

raw format

uncompressed format containing raw pixel data (e.g. YUV, RGB formats).

resume point

a point in the bytestream from which decoding may start/continue, without any previous state/data present, e.g.: a keyframe (VP8/VP9) or SPS/PPS/IDR sequence (H.264/HEVC); a resume point is required to start decode of a new stream, or to resume decoding after a seek.

source

data fed to the decoder or encoder; see `OUTPUT`.

source height

height in pixels for given source resolution; relevant to encoders only.

source resolution

resolution in pixels of source frames being source to the encoder and subject to further cropping to the bounds of visible resolution; relevant to encoders only.

source width

width in pixels for given source resolution; relevant to encoders only.

SPS

Sequence Parameter Set; a type of metadata entity in an H.264/HEVC bytestream.

stream metadata

additional (non-visual) information contained inside encoded bytestream; for example: coded resolution, visible resolution, codec profile.

visible height

height for given visible resolution; display height.

visible resolution

stream resolution of the visible picture, in pixels, to be used for display purposes; must be smaller or equal to coded resolution; display resolution.

visible width

width for given visible resolution; display width.

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-decoder.rst, line 163)

Unknown directive type "kernel-render".

```
.. kernel-render:: DOT
   :alt: DOT digraph of decoder state machine
   :caption: Decoder State Machine

   digraph decoder_state_machine {
       node [shape = doublecircle, label="Decoding"] Decoding;

       node [shape = circle, label="Initialization"] Initialization;
       node [shape = circle, label="Capture\nsetup"] CaptureSetup;
       node [shape = circle, label="Dynamic\nResolution\nChange"] ResChange;
       node [shape = circle, label="Stopped"] Stopped;
       node [shape = circle, label="Drain"] Drain;
       node [shape = circle, label="Seek"] Seek;
       node [shape = circle, label="End of Stream"] EoS;

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

       Initialization -> CaptureSetup [ label = "CAPTURE\nformat\nestablished" ];

       CaptureSetup -> Stopped [ label = "CAPTURE\nbuffers\nready" ];

       Decoding -> ResChange [ label = "Stream\nresolution\nchange" ];
       Decoding -> Drain [ label = "V4L2_DEC_CMD_STOP" ];
       Decoding -> EoS [ label = "EoS mark\nin the stream" ];
       Decoding -> Seek [ label = "VIDIOC_STREAMOFF(OUTPUT)" ];
       Decoding -> Stopped [ label = "VIDIOC_STREAMOFF(CAPTURE)" ];
       Decoding -> Decoding;

       ResChange -> CaptureSetup [ label = "CAPTURE\nformat\nestablished" ];
       ResChange -> Seek [ label = "VIDIOC_STREAMOFF(OUTPUT)" ];

       EoS -> Drain [ label = "Implicit\ndrain" ];

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

       Seek -> Decoding [ label = "VIDIOC_STREAMON(OUTPUT)" ];
       Seek -> Initialization [ label = "VIDIOC_REQBUFS(OUTPUT, 0)" ];

       Stopped -> Decoding [ label = "V4L2_DEC_CMD_START\nor\nVIDIOC_STREAMON(CAPTURE)" ];
       Stopped -> Seek [ label = "VIDIOC_STREAMOFF(OUTPUT)" ];
   }
```

Querying Capabilities

1. To enumerate the set of coded formats supported by the decoder, 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-decoder.rst, line 210); *backlink*

   Unknown interpreted text role "c:func".

   - The full set of supported formats will be returned, regardless of the format set on `CAPTURE`.
   - Check the flags field of :c:type:`v4l2_fmtdesc` for more information about the decoder's capabilities with respect to each coded format. In particular whether or not the decoder has a full-fledged bytestream parser and if the decoder supports dynamic resolution changes.

     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-decoder.rst, line 215); *backlink*

     Unknown interpreted text role "c:type".

2. To enumerate the set of supported raw formats, 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-decoder.rst, line 220); *backlink*
>
> Unknown interpreted text role "c:func".

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

3. The client may use :c:func:`VIDIOC_ENUM_FRAMESIZES` to detect supported resolutions for a given format, passing 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-decoder.rst, line 230); *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-decoder.rst, line 230); *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 decoder for 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-decoder.rst, line 234); *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 decoder for given raw pixel format and the coded format currently set 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-decoder.rst, line 238); *backlink*
>
> Unknown interpreted text role "c:func".

4. Supported profiles and levels for the coded format currently set on `OUTPUT`, 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-decoder.rst, line 243); *backlink*
>
> Unknown interpreted text role "c:func".

Initialization

1. Set the coded format on `OUTPUT` 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-decoder.rst, line 250); *backlink*
>
> Unknown interpreted text role "c:func".

- **Required fields:**

  `type`

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

  `pixelformat`

a coded pixel format.

`width`, `height`
: coded resolution of the stream; required only if it cannot be parsed from the stream for the given coded format; otherwise the decoder will use this resolution as a placeholder resolution that will likely change as soon as it can parse the actual coded resolution from the stream.

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

other fields
: follow standard semantics.

- **Return fields:**

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

- The `CAPTURE` format will be updated with an appropriate frame buffer resolution instantly based on the width and height returned by :c:func:`VIDIOC_S_FMT`. However, for coded formats that include stream resolution information, after the decoder is done parsing the information from the stream, it will update the `CAPTURE` format with new values and signal a source change event, regardless of whether they match the values set by the client or not.

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-decoder.rst`, **line 278**); ***backlink***

Unknown interpreted text role "c:func".

Important

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

2. Allocate source (bytestream) buffers via :c:func:`VIDIOC_REQBUFS` 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-decoder.rst`, **line 293**); ***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`.

  `memory`
  : follows standard semantics.

- **Return fields:**

  `count`
  : the 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.

Alternatively, :c:func:`VIDIOC_CREATE_BUFS` on the `OUTPUT` queue 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-decoder.rst`, **line 318**); ***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`.

`memory`
: follows standard semantics.

`format`
: follows standard semantics.

- **Return fields:**

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

> **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.

3. Start streaming on the `OUTPUT` queue via :c:func:`VIDIOC_STREAMON`.

> **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-decoder.rst`, **line 346**); ***backlink***
>
> Unknown interpreted text role "c:func".

4. **This step only applies to coded formats that contain resolution information in the stream.** Continue queuing/dequeuing bytestream buffers to/from the `OUTPUT` queue via :c:func:`VIDIOC_QBUF` and :c:func:`VIDIOC_DQBUF`. The buffers will be processed and returned to the client in order, until required metadata to configure the `CAPTURE` queue are found. This is indicated by the decoder sending a `V4L2_EVENT_SOURCE_CHANGE` event with `changes` set to `V4L2_EVENT_SRC_CH_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-decoder.rst`, **line 348**); ***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-decoder.rst`, **line 348**); ***backlink***
>
> Unknown interpreted text role "c:func".

- It is not an error if the first buffer does not contain enough data for this to occur. Processing of the buffers will continue as long as more data is needed.
- If data in a buffer that triggers the event is required to decode the first frame, it will not be returned to the client, until the initialization sequence completes and the frame is decoded.
- If the client has not set the coded resolution of the stream on its own, calling :c:func:`VIDIOC_G_FMT`, :c:func:`VIDIOC_S_FMT`, :c:func:`VIDIOC_TRY_FMT` or :c:func:`VIDIOC_REQBUFS` on the `CAPTURE` queue will not return the real values for the stream until a `V4L2_EVENT_SOURCE_CHANGE` event with `changes` set to `V4L2_EVENT_SRC_CH_RESOLUTION` is signaled.

  > **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-decoder.rst`, **line 364**); ***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-decoder.rst`, **line 364**); ***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-decoder.rst`, **line 364**); ***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-decoder.rst`, line 364); *backlink*

Unknown interpreted text role "c:func".

Important

Any client query issued after the decoder queues the event will return values applying to the just parsed stream, including queue formats, selection rectangles and controls.

Note

A client capable of acquiring stream parameters from the bytestream on its own may attempt to set the width and height of the `OUTPUT` format to non-zero values matching the coded size of the stream, skip this step and continue with the *Capture Setup* sequence. However, it must not rely on any driver queries regarding stream parameters, such as selection rectangles and controls, since the decoder has not parsed them from the stream yet. If the values configured by the client do not match those parsed by the decoder, a *Dynamic Resolution Change* will be triggered to reconfigure them.

Note

No decoded frames are produced during this phase.

5. Continue with the *Capture Setup* sequence.

Capture Setup

1. Call :c:func:`VIDIOC_G_FMT` on the `CAPTURE` queue to get format for the destination buffers parsed/decoded from the bytestream.

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-decoder.rst`, line 398); *backlink*

Unknown interpreted text role "c:func".

- **Required fields:**

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

- **Return fields:**

  `width`, `height`
  : frame buffer resolution for the decoded frames.

  `pixelformat`
  : pixel format for decoded frames.

  `num_planes` (for _MPLANE `type` only)
  : number of planes for pixelformat.

  `sizeimage`, `bytesperline`
  : as per standard semantics; matching frame buffer format.

Note

The value of `pixelformat` may be any pixel format supported by the decoder for the current stream. The decoder should choose a preferred/optimal format for the default configuration. For example, a YUV format may be preferred over an RGB format if an additional conversion step would be required for the latter.

2. **Optional.** Acquire the visible resolution via :c:func:`VIDIOC_G_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-decoder.rst`, line 428); *backlink*

Unknown interpreted text role "c:func".

- **Required fields:**

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

  `target`
  : set to `V4L2_SEL_TGT_COMPOSE`.

- **Return fields:**

  `r.left`, `r.top`, `r.width`, `r.height`
  : the visible rectangle; it must fit within the frame buffer resolution returned by :c:func:`VIDIOC_G_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-decoder.rst`, line 442); *backlink*

  Unknown interpreted text role "c:func".

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

  `V4L2_SEL_TGT_CROP_BOUNDS`
  : corresponds to the coded resolution of the stream.

  `V4L2_SEL_TGT_CROP_DEFAULT`
  : the rectangle covering the part of the `CAPTURE` buffer that contains meaningful picture data (visible area); width and height will be equal to the visible resolution of the stream.

  `V4L2_SEL_TGT_CROP`
  : the rectangle within the coded resolution to be output to `CAPTURE`; defaults to `V4L2_SEL_TGT_CROP_DEFAULT`; read-only on hardware without additional compose/scaling capabilities.

  `V4L2_SEL_TGT_COMPOSE_BOUNDS`
  : the maximum rectangle within a `CAPTURE` buffer, which the cropped frame can be composed into; equal to `V4L2_SEL_TGT_CROP` if the hardware does not support compose/scaling.

  `V4L2_SEL_TGT_COMPOSE_DEFAULT`
  : equal to `V4L2_SEL_TGT_CROP`.

  `V4L2_SEL_TGT_COMPOSE`
  : the rectangle inside a `CAPTURE` buffer into which the cropped frame is written; defaults to `V4L2_SEL_TGT_COMPOSE_DEFAULT`; read-only on hardware without additional compose/scaling capabilities.

  `V4L2_SEL_TGT_COMPOSE_PADDED`
  : the rectangle inside a `CAPTURE` buffer which is overwritten by the hardware; equal to `V4L2_SEL_TGT_COMPOSE` if the hardware does not write padding pixels.

Warning

The values are guaranteed to be meaningful only after the decoder successfully parses the stream metadata. The client must not rely on the query before that happens.

3. **Optional.** Enumerate `CAPTURE` formats via :c:func:`VIDIOC_ENUM_FMT` on the `CAPTURE` queue. Once the stream information is parsed and known, the client may use this ioctl to discover which raw formats are supported for given stream and select one of them 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-decoder.rst`, line 484); *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-decoder.rst`, line 484); *backlink*

   Unknown interpreted text role "c:func".

   Important

The decoder will return only formats supported for the currently established coded format, as per the OUTPUT format and/or stream metadata parsed in this initialization sequence, even if more formats may be supported by the decoder in general. In other words, the set returned will be a subset of the initial query mentioned in the *Querying Capabilities* section.

For example, a decoder may support YUV and RGB formats for resolutions 1920x1088 and lower, but only YUV for higher resolutions (due to hardware limitations). After parsing a resolution of 1920x1088 or lower, :c:func:`VIDIOC_ENUM_FMT` may return a set of YUV and RGB pixel formats, but after parsing resolution higher than 1920x1088, the decoder will not return RGB, unsupported for this 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-decoder.rst, line 498); *backlink*

Unknown interpreted text role "c:func".

However, subsequent resolution change event triggered after discovering a resolution change within the same stream may switch the stream into a lower resolution and :c:func:`VIDIOC_ENUM_FMT` would return RGB formats again in that case.

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-decoder.rst, line 505); *backlink*

Unknown interpreted text role "c:func".

4. **Optional.** Set the CAPTURE format via :c:func:`VIDIOC_S_FMT` on the CAPTURE queue. The client may choose a different format than selected/suggested by the decoder in :c:func:`VIDIOC_G_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-decoder.rst, line 510); *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-decoder.rst, line 510); *backlink*

Unknown interpreted text role "c:func".

- **Required fields:**

  type

  a V4L2_BUF_TYPE_* enum appropriate for CAPTURE.

  pixelformat

  a raw pixel format.

  width, height

  frame buffer resolution of the decoded stream; typically unchanged from what was returned with :c:func:`VIDIOC_G_FMT`, but it may be different if the hardware supports composition and/or scaling.

  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-decoder.rst, line 523); *backlink*

  Unknown interpreted text role "c:func".

- Setting the CAPTURE format will reset the compose selection rectangles to their default values, based on the new resolution, as described in the previous step.

5. **Optional.** Set the compose rectangle via :c:func:`VIDIOC_S_SELECTION` on the CAPTURE queue if it is desired and if the decoder has compose and/or scaling capabilities.

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-decoder.rst`, line 531); *backlink*

Unknown interpreted text role "c:func".

- **Required fields:**

  `type`

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

  `target`

  set to `V4L2_SEL_TGT_COMPOSE`.

  `r.left`, `r.top`, `r.width`, `r.height`

  the rectangle inside a `CAPTURE` buffer into which the cropped frame is written; defaults to `V4L2_SEL_TGT_COMPOSE_DEFAULT`; read-only on hardware without additional compose/scaling capabilities.

- **Return fields:**

  `r.left`, `r.top`, `r.width`, `r.height`

  the visible rectangle; it must fit within the frame buffer resolution returned by :c:func:`VIDIOC_G_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-decoder.rst`, line 551); *backlink*

  Unknown interpreted text role "c:func".

Warning

The decoder may adjust the compose rectangle to the nearest supported one 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-decoder.rst`, line 556); *backlink*

Unknown interpreted text role "c:func".

6. If all the following conditions are met, the client may resume the decoding instantly:

   - `sizeimage` of the new format (determined in previous steps) is less than or equal to the size of currently allocated buffers,
   - the number of buffers currently allocated is greater than or equal to the minimum number of buffers acquired in previous steps. To fulfill this requirement, the client may use :c:func:`VIDIOC_CREATE_BUFS` to add new buffers.

   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-decoder.rst`, line 566); *backlink*

   Unknown interpreted text role "c:func".

   In that case, the remaining steps do not apply and the client may resume the decoding by one of the following actions:

   - if the `CAPTURE` queue is streaming, call :c:func:`VIDIOC_DECODER_CMD` with the `V4L2_DEC_CMD_START` command,

   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-decoder.rst`, line 574); *backlink*

   Unknown interpreted text role "c:func".

   - if the `CAPTURE` queue is not streaming, call :c:func:`VIDIOC_STREAMON` on the `CAPTURE` queue.

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-decoder.rst`, line 577); *backlink*

Unknown interpreted text role "c:func".

However, if the client intends to change the buffer set, to lower memory usage or for any other reasons, it may be achieved by following the steps below.

7. **If the** `CAPTURE` **queue is streaming,** keep queuing and dequeuing buffers on the `CAPTURE` queue until a buffer marked with the `V4L2_BUF_FLAG_LAST` flag is dequeued.

8. **If the** `CAPTURE` **queue is streaming,** call :c:func:`VIDIOC_STREAMOFF` on the `CAPTURE` queue to stop 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-decoder.rst`, line 588); *backlink*

   Unknown interpreted text role "c:func".

   Warning

   The `OUTPUT` queue must remain streaming. Calling :c:func:`VIDIOC_STREAMOFF` on it would abort the sequence and trigger a seek.

   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-decoder.rst`, line 593); *backlink*

   Unknown interpreted text role "c:func".

9. **If the** `CAPTURE` **queue has buffers allocated,** free the `CAPTURE` buffers using :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-decoder.rst`, line 597); *backlink*

   Unknown interpreted text role "c:func".

   - **Required fields:**

     `count`
     set to 0.

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

     `memory`
     follows standard semantics.

10. Allocate `CAPTURE` buffers via :c:func:`VIDIOC_REQBUFS` on the `CAPTURE` queue.

    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-decoder.rst`, line 611); *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 `CAPTURE`.

      `memory`
      follows 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 buffers (for pipeline depth), the client may query the `V4L2_CID_MIN_BUFFERS_FOR_CAPTURE` 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-decoder.rst`, **line 638**); *backlink*
> >
> > Unknown interpreted text role "c:func".

Alternatively, :c:func:`VIDIOC_CREATE_BUFS` on the `CAPTURE` queue can be used to have more control over buffer allocation. For example, by allocating buffers larger than the current `CAPTURE` format, future resolution changes can be accommodated.

> **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-decoder.rst`, **line 644**); *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 `CAPTURE`.

  `memory`
  : follows standard semantics.

  `format`
  : a format representing the maximum framebuffer resolution to be accommodated by newly allocated buffers.

- **Return fields:**

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

> **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 buffers for a format different than parsed from the stream metadata, the client must proceed as follows, before the metadata parsing is initiated:
>
> - set width and height of the `OUTPUT` format to desired coded resolution to let the decoder configure the `CAPTURE` format appropriately,
> - query the `CAPTURE` format using :c:func:`VIDIOC_G_FMT` and save it until this step.
>
> > **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-decoder.rst`, **line 684**); *backlink*
> >
> > Unknown interpreted text role "c:func".
>
> The format obtained in the query may be then used with :c:func:`VIDIOC_CREATE_BUFS` in this step to allocate the buffers.

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-decoder.rst`, line 687); *backlink*

Unknown interpreted text role "c:func".

11. Call :c:func:`VIDIOC_STREAMON` on the `CAPTURE` queue to start decoding frames.

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-decoder.rst`, line 690); *backlink*

Unknown interpreted text role "c:func".

Decoding

This state is reached after the *Capture Setup* 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-decoder.rst`, line 696); *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-decoder.rst`, line 696); *backlink*

Unknown interpreted text role "c:func".

The content of the source `OUTPUT` 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 the standard behavior of V4L2 buffer queues and memory-to-memory devices. In addition, the order of decoded frames dequeued from the `CAPTURE` queue may differ from the order of queuing coded 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 no buffers being produced on `CAPTURE` (e.g. if it does not contain encoded data, or if only metadata syntax structures are present in it),
- a buffer queued to `OUTPUT` may result in more than one buffer produced on `CAPTURE` (if the encoded data contained more than one frame, or if returning a decoded frame allowed the decoder to return a frame that preceded it in decode, but succeeded it in the display order),
- a buffer queued to `OUTPUT` may result in a buffer being produced on `CAPTURE` later into decode 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 decoding results are only available at later time, due to specifics of the decoding process.

Note

To allow matching decoded `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 decoding 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-decoder.rst`, line 736); *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.
- multiple OUTPUT buffers generate one CAPTURE buffer: timestamp of the OUTPUT buffer queued first will be copied.
- the decoding order differs from the display 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

The backing memory of CAPTURE buffers that are used as reference frames by the stream may be read by the hardware even after they are dequeued. Consequently, the client should avoid writing into this memory while the CAPTURE queue is streaming. Failure to observe this may result in corruption of decoded frames.

Similarly, when using a memory type other than V4L2_MEMORY_MMAP, the client should make sure that each CAPTURE buffer is always queued with the same backing memory for as long as the CAPTURE queue is streaming. The reason for this is that V4L2 buffer indices can be used by drivers to identify frames. Thus, if the backing memory of a reference frame is submitted under a different buffer ID, the driver may misidentify it and decode a new frame into it while it is still in use, resulting in corruption of the following frames.

During the decoding, the decoder may initiate one of the special sequences, as listed below. The sequences will result in the decoder returning all the CAPTURE buffers that originated from all the OUTPUT buffers processed before the sequence started. Last of the buffers will have the V4L2_BUF_FLAG_LAST flag set. To determine the sequence to follow, the client must check if there is any pending event and:

- if a V4L2_EVENT_SOURCE_CHANGE event with changes set to V4L2_EVENT_SRC_CH_RESOLUTION is pending, the *Dynamic Resolution Change* sequence needs to be followed,
- if a V4L2_EVENT_EOS event is pending, the *End of Stream* sequence needs to be followed.

Some of the sequences can be intermixed with each other and need to be handled as they happen. The exact operation is documented for each sequence.

Should a decoding error occur, it will be reported to the client with the level of details depending on the decoder capabilities. Specifically:

- the CAPTURE buffer that contains the results of the failed decode operation will be returned with the V4L2_BUF_FLAG_ERROR flag set,
- if the decoder is able to precisely report the OUTPUT buffer that triggered the error, such buffer will be returned with the V4L2_BUF_FLAG_ERROR flag set.

In case of a fatal failure that does not allow the decoding to continue, any further operations on corresponding decoder 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.

Seek

Seek is controlled by the OUTPUT queue, as it is the source of coded data. The seek does not require any specific operation on the CAPTURE queue, but it may be affected as per normal decoder operation.

1. Stop the OUTPUT queue to begin the seek sequence via :c:func:`VIDIOC_STREAMOFF`.

   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-decoder.rst, line 812); *backlink*

   Unknown interpreted text role "c:func".

   - **Required fields:**

     type

     a V4L2_BUF_TYPE_* enum appropriate for OUTPUT.

   - The decoder will drop all the pending OUTPUT buffers and they must be treated as returned to the client (following standard semantics).

2. Restart the OUTPUT queue via :c:func:`VIDIOC_STREAMON`.

   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-decoder.rst, line 823); *backlink*

   Unknown interpreted text role "c:func".

- **Required fields:**

  `type`

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

- The decoder will start accepting new source bytestream buffers after the call returns.

3. Start queuing buffers containing coded data after the seek to the `OUTPUT` queue until a suitable resume point is found.

> **Note**
>
> There is no requirement to begin queuing coded data starting exactly from a resume point (e.g. SPS or a keyframe). Any queued `OUTPUT` buffers will be processed and returned to the client until a suitable resume point is found. While looking for a resume point, the decoder should not produce any decoded frames into `CAPTURE` buffers.
>
> Some hardware is known to mishandle seeks to a non-resume point. Such an operation may result in an unspecified number of corrupted decoded frames being made available on the `CAPTURE` queue. Drivers must ensure that no fatal decoding errors or crashes occur, and implement any necessary handling and workarounds for hardware issues related to seek operations.

> **Warning**
>
> In case of the H.264/HEVC codec, the client must take care not to seek over a change of SPS/PPS. Even though the target frame could be a keyframe, the stale SPS/PPS inside decoder state would lead to undefined results when decoding. Although the decoder must handle that case without a crash or a fatal decode error, the client must not expect a sensible decode output.
>
> If the hardware can detect such corrupted decoded frames, then corresponding buffers will be returned to the client with the V4L2_BUF_FLAG_ERROR set. See the *Decoding* section for further description of decode error reporting.

4. After a resume point is found, the decoder will start returning `CAPTURE` buffers containing decoded frames.

> **Important**
>
> A seek may result in the *Dynamic Resolution Change* sequence being initiated, due to the seek target having decoding parameters different from the part of the stream decoded before the seek. The sequence must be handled as per normal decoder operation.

> **Warning**
>
> It is not specified when the `CAPTURE` queue starts producing buffers containing decoded data from the `OUTPUT` buffers queued after the seek, as it operates independently from the `OUTPUT` queue.
>
> The decoder may return a number of remaining `CAPTURE` buffers containing decoded frames originating from the `OUTPUT` buffers queued before the seek sequence is performed.
>
> The `VIDIOC_STREAMOFF` operation discards any remaining queued `OUTPUT` buffers, which means that not all of the `OUTPUT` buffers queued before the seek sequence may have matching `CAPTURE` buffers produced. For example, given the sequence of operations on the `OUTPUT` queue:
>
> QBUF(A), QBUF(B), STREAMOFF(), STREAMON(), QBUF(G), QBUF(H),
>
> any of the following results on the `CAPTURE` queue is allowed:
>
> {A', B', G', H'}, {A', G', H'}, {G', H'}.
>
> To determine the CAPTURE buffer containing the first decoded frame after the seek, the client may observe the timestamps to match the CAPTURE and OUTPUT buffers or use V4L2_DEC_CMD_STOP and V4L2_DEC_CMD_START to drain the decoder.

> **Note**
>
> To achieve instantaneous seek, the client may restart streaming on the `CAPTURE` queue too to discard decoded, but not yet dequeued buffers.

Dynamic Resolution Change

Streams that include resolution metadata in the bytestream may require switching to a different resolution during the decoding.

Note

Not all decoders can detect resolution changes. Those that do set the `V4L2_FMT_FLAG_DYN_RESOLUTION` flag for the coded format when :c:func:`VIDIOC_ENUM_FMT` is called.

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-decoder.rst`, **line 914**); ***backlink***

Unknown interpreted text role "c:func".

The sequence starts when the decoder detects a coded frame with one or more of the following parameters different from those previously established (and reflected by corresponding queries):

- coded resolution (`OUTPUT` width and height),
- visible resolution (selection rectangles),
- the minimum number of buffers needed for decoding,
- bit-depth of the bitstream has been changed.

Whenever that happens, the decoder must proceed as follows:

1. After encountering a resolution change in the stream, the decoder sends a `V4L2_EVENT_SOURCE_CHANGE` event with `changes` set to `V4L2_EVENT_SRC_CH_RESOLUTION`.

   Important

   Any client query issued after the decoder queues the event will return values applying to the stream after the resolution change, including queue formats, selection rectangles and controls.

2. The decoder will then process and decode all remaining buffers from before the resolution change point.

   - The last buffer from before the change must be marked with the `V4L2_BUF_FLAG_LAST` flag, similarly to the *Drain* sequence above.

   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 a decoded 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-decoder.rst`, **line 950**); ***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-decoder.rst`, **line 956**); ***backlink***

   Unknown interpreted text role "c:func".

The client must continue the sequence as described below to continue the decoding process.

1. Dequeue the source change event.

   Important

   A source change triggers an implicit decoder drain, similar to the explicit *Drain* sequence. The decoder is stopped after it completes. The decoding process must be resumed with either a pair of calls to :c:func:`VIDIOC_STREAMOFF` and :c:func:`VIDIOC_STREAMON` on the `CAPTURE` queue, or a call to :c:func:`VIDIOC_DECODER_CMD` with the `V4L2_DEC_CMD_START` command.

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-decoder.rst`, line 967); *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-decoder.rst`, line 967); *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-decoder.rst`, line 967); *backlink*

Unknown interpreted text role "c:func".

2. Continue with the *Capture Setup* sequence.

Note

During the resolution change sequence, the `OUTPUT` queue must remain streaming. Calling :c:func:`VIDIOC_STREAMOFF` on the `OUTPUT` queue would abort the sequence and initiate a seek.

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-decoder.rst`, line 978); *backlink*

Unknown interpreted text role "c:func".

In principle, the `OUTPUT` queue operates separately from the `CAPTURE` queue and this remains true for the duration of the entire resolution change sequence as well.

The client should, for best performance and simplicity, keep queuing/dequeuing buffers to/from the `OUTPUT` queue even while processing this sequence.

Drain

To ensure that all queued `OUTPUT` buffers have been processed and 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 decoded frames for all `OUTPUT` buffers queued before the sequence was started.

1. Begin drain by issuing :c:func:`VIDIOC_DECODER_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-decoder.rst`, line 998); *backlink*

Unknown interpreted text role "c:func".

- **Required fields:**

  `cmd`
    set to `V4L2_DEC_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_DECODER_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-decoder.rst, line 1013); *backlink***
>
> Unknown interpreted text role "c:func".

2. Any OUTPUT buffers queued by the client before the :c:func:`VIDIOC_DECODER_CMD` was issued will be processed and decoded as normal. The client must continue to handle both queues independently, similarly to normal decode 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-decoder.rst, line 1019); *backlink***
>
> Unknown interpreted text role "c:func".

- handling any operations triggered as a result of processing those buffers, such as the *Dynamic Resolution Change* sequence, before continuing with the drain sequence,
- 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 a decoded 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-decoder.rst, line 1033); *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-decoder.rst, line 1039); *backlink***
> >
> > Unknown interpreted text role "c:func".

- dequeuing processed OUTPUT buffers, until all the buffers queued before the V4L2_DEC_CMD_STOP command are dequeued,
- dequeuing the V4L2_EVENT_EOS event, if the client subscribed to it.

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

3. Once all the OUTPUT buffers queued before the V4L2_DEC_CMD_STOP call are dequeued and the last CAPTURE buffer is dequeued, the decoder is stopped and it will accept, but not process, any newly queued OUTPUT buffers until the client issues any of the following operations:

- `V4L2_DEC_CMD_START` - the decoder 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 decoder will resume the operation normally, however any `CAPTURE` buffers still in the queue will be 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-decoder.rst`, line 1063); ***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-decoder.rst`, line 1063); ***backlink***

  Unknown interpreted text role "c:func".

- a pair of :c:func:`VIDIOC_STREAMOFF` and :c:func:`VIDIOC_STREAMON` on the `OUTPUT` queue - any pending source buffers will be returned to the client and the *Seek* sequence will be triggered.

  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-decoder.rst`, line 1068); ***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-decoder.rst`, line 1068); ***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_DEC_CMD_START` or `V4L2_DEC_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-decoder.rst`, line 1074); ***backlink***

Unknown interpreted text role "c:func".

Although not mandatory, the availability of decoder commands may be queried using :c:func:`VIDIOC_TRY_DECODER_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-decoder.rst`, line 1081); ***backlink***

Unknown interpreted text role "c:func".

End of Stream

If the decoder encounters an end of stream marking in the stream, the decoder will initiate the *Drain* sequence, which the client must handle as described above, skipping the initial :c:func:`VIDIOC_DECODER_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-decoder.rst`, line 1087); ***backlink***

Unknown interpreted text role "c:func".

Commit Points

Setting formats and allocating buffers trigger changes in the behavior of the decoder.

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