ioctl VIDIOC_DECODER_CMD, 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)vidioc-decoder-cmd.rst, line 2)

Unknown directive type "c:namespace".

```
.. c:namespace:: V4L
```

Name

VIDIOC_DECODER_CMD - VIDIOC_TRY_DECODER_CMD - Execute an decoder command

Synopsis

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 18)

Unknown directive type "c:macro".

```
.. c:macro:: VIDIOC_DECODER_CMD
```

```
int ioctl(int fd, VIDIOC_DECODER_CMD, struct v4l2_decoder_cmd *argp)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 22)

Unknown directive type "c:macro".

```
.. c:macro:: VIDIOC_TRY_DECODER_CMD
```

```
int ioctl(int fd, VIDIOC_TRY_DECODER_CMD, struct v4l2_decoder_cmd *argp)
```

Arguments

`fd`

File descriptor returned by :c:func:`open()`.

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

Unknown interpreted text role "c:func".

`argp`

pointer to struct :c:type:`v4l2_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)vidioc-decoder-cmd.rst, line 33); *backlink*

Unknown interpreted text role "c:type".

Description

These ioctls control an audio/video (usually MPEG-) decoder. `VIDIOC_DECODER_CMD` sends a command to the decoder,

VIDIOC_TRY_DECODER_CMD can be used to try a command without actually executing it. To send a command applications must initialize all fields of a struct :c:type:`v4l2_decoder_cmd` and call VIDIOC_DECODER_CMD or VIDIOC_TRY_DECODER_CMD with a pointer to this structure.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 38); *backlink*

Unknown interpreted text role "c:type".

The cmd field must contain the command code. Some commands use the flags field for additional information.

A :c:func:`write()` or :ref:`VIDIOC_STREAMON` call sends an implicit START command to the decoder if it has not been started yet. Applies to both queues of mem2mem decoders.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 49); *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)vidioc-decoder-cmd.rst, line 49); *backlink*

Unknown interpreted text role "ref".

A :c:func:`close()` or :ref:`VIDIOC_STREAMOFF <VIDIOC_STREAMON>` call of a streaming file descriptor sends an implicit immediate STOP command to the decoder, and all buffered data is discarded. Applies to both queues of mem2mem decoders.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 53); *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)vidioc-decoder-cmd.rst, line 53); *backlink*

Unknown interpreted text role "ref".

In principle, these ioctls are optional, not all drivers may support them. They were introduced in Linux 3.3. They are, however, mandatory for stateful mem2mem decoders (as further documented in :ref:`decoder`).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 58); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 62)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{2.0cm}|p{1.1cm}|p{2.2cm}|p{11.8cm}|
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 64)

Unknown directive type "c:type".

```
.. c:type:: v4l2_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)vidioc-decoder-cmd.rst, line 66)

Unknown directive type "cssclass".

```
.. cssclass:: longtable
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 68)

Unknown directive type "flat-table".

```
.. flat-table:: struct v4l2_decoder_cmd
    :header-rows:  0
    :stub-columns: 0
    :widths: 1 1 1 3

    * - __u32
      - ``cmd``
      -
      - The decoder command, see :ref:`decoder-cmds`.
    * - __u32
      - ``flags``
      -
      - Flags to go with the command. If no flags are defined for this
        command, drivers and applications must set this field to zero.
    * - union {
      - (anonymous)
    * - struct
      - ``start``
      -
      - Structure containing additional data for the
        ``V4L2_DEC_CMD_START`` command.
    * -
      - __s32
      - ``speed``
      - Playback speed and direction. The playback speed is defined as
        ``speed``/1000 of the normal speed. So 1000 is normal playback.
        Negative numbers denote reverse playback, so -1000 does reverse
        playback at normal speed. Speeds -1, 0 and 1 have special
        meanings: speed 0 is shorthand for 1000 (normal playback). A speed
        of 1 steps just one frame forward, a speed of -1 steps just one
        frame back.
    * -
      - __u32
      - ``format``
      - Format restrictions. This field is set by the driver, not the
        application. Possible values are ``V4L2_DEC_START_FMT_NONE`` if
        there are no format restrictions or ``V4L2_DEC_START_FMT_GOP`` if
        the decoder operates on full GOPs (*Group Of Pictures*). This is
        usually the case for reverse playback: the decoder needs full
        GOPs, which it can then play in reverse order. So to implement
        reverse playback the application must feed the decoder the last
        GOP in the video file, then the GOP before that, etc. etc.
    * - struct
      - ``stop``
      -
      - Structure containing additional data for the ``V4L2_DEC_CMD_STOP``
        command.
    * -
      - __u64
      - ``pts``
      - Stop playback at this ``pts`` or immediately if the playback is
        already past that timestamp. Leave to 0 if you want to stop after
        the last frame was decoded.
    * - struct
      - ``raw``
    * -
      - __u32
      - ``data``\ [16]
      - Reserved for future extensions. Drivers and applications must set
        the array to zero.
    * - }
      -
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 132)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{5.6cm}|p{0.6cm}|p{11.1cm}|
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 134)

Unknown directive type "cssclass".

```
.. cssclass:: longtable
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 138)

Unknown directive type "flat-table".

```
.. flat-table:: Decoder Commands
    :header-rows:  0
    :stub-columns: 0
    :widths: 56 6 113

    * - ``V4L2_DEC_CMD_START``
      - 0
      - Start the decoder. When the decoder is already running or paused,
        this command will just change the playback speed. That means that
        calling ``V4L2_DEC_CMD_START`` when the decoder was paused will
        *not* resume the decoder. You have to explicitly call
        ``V4L2_DEC_CMD_RESUME`` for that. This command has one flag:
        ``V4L2_DEC_CMD_START_MUTE_AUDIO``. If set, then audio will be
        muted when playing back at a non-standard speed.

        For a device implementing the :ref:`decoder`, once the drain sequence
        is initiated with the ``V4L2_DEC_CMD_STOP`` command, it must be driven
        to completion before this command can be invoked.  Any attempt to
        invoke the command while the drain sequence is in progress will trigger
        an ``EBUSY`` error code.  The command may be also used to restart the
        decoder in case of an implicit stop initiated by the decoder itself,
        without the ``V4L2_DEC_CMD_STOP`` being called explicitly. See
        :ref:`decoder` for more details.
    * - ``V4L2_DEC_CMD_STOP``
      - 1
      - Stop the decoder. When the decoder is already stopped, this
        command does nothing. This command has two flags: if
        ``V4L2_DEC_CMD_STOP_TO_BLACK`` is set, then the decoder will set
        the picture to black after it stopped decoding. Otherwise the last
        image will repeat. If
        ``V4L2_DEC_CMD_STOP_IMMEDIATELY`` is set, then the decoder stops
        immediately (ignoring the ``pts`` value), otherwise it will keep
        decoding until timestamp >= pts or until the last of the pending
        data from its internal buffers was decoded.

        For a device implementing the :ref:`decoder`, the command will initiate
        the drain sequence as documented in :ref:`decoder`.  No flags or other
        arguments are accepted in this case. Any attempt to invoke the command
        again before the sequence completes will trigger an ``EBUSY`` error
        code.
    * - ``V4L2_DEC_CMD_PAUSE``
      - 2
      - Pause the decoder. When the decoder has not been started yet, the
        driver will return an ``EPERM`` error code. When the decoder is
        already paused, this command does nothing. This command has one
        flag: if ``V4L2_DEC_CMD_PAUSE_TO_BLACK`` is set, then set the
        decoder output to black when paused.
    * - ``V4L2_DEC_CMD_RESUME``
      - 3
      - Resume decoding after a PAUSE command. When the decoder has not
        been started yet, the driver will return an ``EPERM`` error code. When
        the decoder is already running, this command does nothing. No
```

```
      flags are defined for this command.
    * - ``V4L2_DEC_CMD_FLUSH``
      - 4
      - Flush any held capture buffers. Only valid for stateless decoders.
        This command is typically used when the application reached the
        end of the stream and the last output buffer had the
        ``V4L2_BUF_FLAG_M2M_HOLD_CAPTURE_BUF`` flag set. This would prevent
        dequeueing the capture buffer containing the last decoded frame.
        So this command can be used to explicitly flush that final decoded
        frame. This command does nothing if there are no held capture buffers.
```

Return Value

On success 0 is returned, on error -1 and the `errno` variable is set appropriately. The generic error codes are described at the :ref:`Generic Error Codes <gen-errors>` chapter.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 204); *backlink*

Unknown interpreted text role "ref".

EBUSY

A drain sequence of a device implementing the :ref:`decoder` is still in progress. It is not allowed to issue another decoder command until it completes.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\(linux-master)(Documentation)(userspace-api)(media)(v4l)vidioc-decoder-cmd.rst, line 209); *backlink*

Unknown interpreted text role "ref".

EINVAL

The `cmd` field is invalid.

EPERM

The application sent a PAUSE or RESUME command when the decoder was not running.