

ioctl VIDIOC_G_EXT_CTRL, VIDIOC_S_EXT_CTRL, VIDIOC_TRY_EXT_CTRL

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-g-ext-ctrls.rst, line 2)

Unknown directive type "c:namespace".

.. c:namespace:: V4L

Name

VIDIOC_G_EXT_CTRL - VIDIOC_S_EXT_CTRL - VIDIOC_TRY_EXT_CTRL - Get or set the value of several controls, try control values

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-g-ext-ctrls.rst, line 18)

Unknown directive type "c:macro".

.. c:macro:: VIDIOC_G_EXT_CTRL

```
int ioctl(int fd, VIDIOC_G_EXT_CTRL, struct v4l2_ext_controls *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-g-ext-ctrls.rst, line 22)

Unknown directive type "c:macro".

.. c:macro:: VIDIOC_S_EXT_CTRL

```
int ioctl(int fd, VIDIOC_S_EXT_CTRL, struct v4l2_ext_controls *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-g-ext-ctrls.rst, line 26)

Unknown directive type "c:macro".

.. c:macro:: VIDIOC_TRY_EXT_CTRL

```
int ioctl(int fd, VIDIOC_TRY_EXT_CTRL, struct v4l2_ext_controls *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-g-ext-ctrls.rst, line 34); [backlink](#)

Unknown interpreted text role "c:func".

argp

Pointer to struct `c:type:'v4l2_ext_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\vidioc-g-ext-ctrls.rst, line 37); [backlink](#)

Unknown interpreted text role "c.type".

Description

These ioctls allow the caller to get or set multiple controls atomically. Control IDs are grouped into control classes (see `ref:ctrl-class`) and all controls in the control array must belong to the same control class.

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-g-ext-ctrls.rst, line 42); [backlink](#)

Unknown interpreted text role "ref".

Applications must always fill in the `count`, `which`, `controls` and `reserved` fields of struct `ref:v4l2_ext_controls`, and initialize the struct `ref:v4l2_ext_control` array pointed to by the `controls` fields.

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-g-ext-ctrls.rst, line 47); [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\vidioc-g-ext-ctrls.rst, line 47); [backlink](#)

Unknown interpreted text role "c.type".

To get the current value of a set of controls applications initialize the `id`, `size` and `reserved2` fields of each struct `ref:v4l2_ext_control` and call the `ref:VIDIOC_G_EXT_CTRL<VIDIOC_G_EXT_CTRL>` ioctl. String controls must also set the `string` field. Controls of compound types (`V4L2_CTRL_FLAG_HAS_PAYLOAD` is set) must set the `ptr` field.

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-g-ext-ctrls.rst, line 53); [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\vidioc-g-ext-ctrls.rst, line 53); [backlink](#)

Unknown interpreted text role "ref".

If the `size` is too small to receive the control result (only relevant for pointer-type controls like strings), then the driver will set `size` to a valid value and return an `ENOSPC` error code. You should re-allocate the memory to this new size and try again. For the string type it is possible that the same issue occurs again if the string has grown in the meantime. It is recommended to call `ref:VIDIOC_QUERYCTRL` first and use `maximum+1` as the new `size` value. It is guaranteed that that is sufficient memory.

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-g-ext-ctrls.rst, line 60); [backlink](#)

Unknown interpreted text role "ref".

N-dimensional arrays are set and retrieved row-by-row. You cannot set a partial array, all elements have to be set or retrieved. The total size is calculated as `elems * elem_size`. These values can be obtained by calling `ref:VIDIOC_QUERY_EXT_CTRL<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\vidioc-g-ext-ctrls.rst, line 70); [backlink](#)

Unknown interpreted text role "ref".

To change the value of a set of controls applications initialize the `id`, `size`, `reserved2` and `value/value64/string/ptr` fields of each struct `:ctype:'v4l2_ext_control'` and call the `ref:'VIDIOC_S_EXT_CTRL <VIDIOC_G_EXT_CTRL>'` ioctl. The controls will only be set if *all* control values are valid.

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-g-ext-ctrls.rst, line 75); [backlink](#)

Unknown interpreted text role "ctype".

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-g-ext-ctrls.rst, line 75); [backlink](#)

Unknown interpreted text role "ref".

To check if a set of controls have correct values applications initialize the `id`, `size`, `reserved2` and `value/value64/string/ptr` fields of each struct `:ctype:'v4l2_ext_control'` and call the `ref:'VIDIOC_TRY_EXT_CTRL <VIDIOC_G_EXT_CTRL>'` ioctl. It is up to the driver whether wrong values are automatically adjusted to a valid value or if an error is returned.

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-g-ext-ctrls.rst, line 81); [backlink](#)

Unknown interpreted text role "ctype".

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-g-ext-ctrls.rst, line 81); [backlink](#)

Unknown interpreted text role "ref".

When the `id` or `which` is invalid drivers return an `EINVAL` error code. When the value is out of bounds drivers can choose to take the closest valid value or return an `ERANGE` error code, whatever seems more appropriate. In the first case the new value is set in struct `:ctype:'v4l2_ext_control'`. If the new control value is inappropriate (e.g. the given menu index is not supported by the menu control), then this will also result in an `EINVAL` error code error.

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-g-ext-ctrls.rst, line 89); [backlink](#)

Unknown interpreted text role "ctype".

If `request_fd` is set to a not-yet-queued `ref:'request <media-request-api>'` file descriptor and `which` is set to `V4L2_CTRL_WHICH_REQUEST_VAL`, then the controls are not applied immediately when calling `ref:'VIDIOC_S_EXT_CTRL <VIDIOC_G_EXT_CTRL>'`, but instead are applied by the driver for the buffer associated with the same request. If the device does not support requests, then `EACCES` will be returned. If requests are supported but an invalid request file descriptor is given, then `EINVAL` will be returned.

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-g-ext-ctrls.rst, line 97); [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-g-ext-ctrls.rst, line 97); [backlink](#)

Unknown interpreted text role "ref".

An attempt to call `ref:'VIDIOC_S_EXT_CTRL <VIDIOC_G_EXT_CTRL>'` for a request that has already been queued will result in an `EBUSY` error.

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-g-ext-ctrls.rst, line 106); [backlink](#)

Unknown interpreted text role "ref".

If `request_fd` is specified and `which` is set to `V4L2_CTRL_WHICH_REQUEST_VAL` during a call to `<ref:VIDIOC_G_EXT_CTRL<VIDIOC_G_EXT_CTRL>`, then it will return the values of the controls at the time of request completion. If the request is not yet completed, then this will result in an `EACCES` error.

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-g-ext-ctrls.rst, line 109); [backlink](#)

Unknown interpreted text role "ref".

The driver will only set/get these controls if all control values are correct. This prevents the situation where only some of the controls were set/get. Only low-level errors (e. g. a failed i2c command) can still cause this situation.

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-g-ext-ctrls.rst, line 121)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{6.8cm}|p{4.0cm}|p{6.5cm}|
```

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-g-ext-ctrls.rst, line 123)

Unknown directive type "c:type".

```
.. c:type:: v4l2_ext_control
```

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-g-ext-ctrls.rst, line 129)

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-g-ext-ctrls.rst, line 131)

Unknown directive type "flat-table".

```
.. flat-table:: struct v4l2_ext_control
:header-rows: 0
:stub-columns: 0
:widths:      1 1 2

* - __u32
  - ``id``
  - Identifies the control, set by the application.
* - __u32
  - ``size``
  - The total size in bytes of the payload of this control.
* - :cspan:2 The ``size`` field is normally 0, but for pointer
  controls this should be set to the size of the memory that contains
  the payload or that will receive the payload.
  If <ref:VIDIOC_G_EXT_CTRL<VIDIOC_G_EXT_CTRL> finds that this value
  is less than is required to store the payload result, then it is set
  to a value large enough to store the payload result and ``ENOSPC`` is
  returned.
```

```
.. note::
```

For string controls, this ```size``` field should

```

        not be confused with the length of the string. This field refers
        to the size of the memory that contains the string. The actual
        *length* of the string may well be much smaller.
* - __u32
  - ``reserved2`` [1]
  - Reserved for future extensions. Drivers and applications must set
    the array to zero.
* - union {
  - (anonymous)
* - __s32
  - ``value``
  - New value or current value. Valid if this control is not of type
    ``V4L2_CTRL_TYPE_INTEGER64`` and ``V4L2_CTRL_FLAG_HAS_PAYLOAD`` is
    not set.
* - __s64
  - ``value64``
  - New value or current value. Valid if this control is of type
    ``V4L2_CTRL_TYPE_INTEGER64`` and ``V4L2_CTRL_FLAG_HAS_PAYLOAD`` is
    not set.
* - char *
  - ``string``
  - A pointer to a string. Valid if this control is of type
    ``V4L2_CTRL_TYPE_STRING``.
* - __u8 *
  - ``p_u8``
  - A pointer to a matrix control of unsigned 8-bit values. Valid if
    this control is of type ``V4L2_CTRL_TYPE_U8``.
* - __u16 *
  - ``p_u16``
  - A pointer to a matrix control of unsigned 16-bit values. Valid if
    this control is of type ``V4L2_CTRL_TYPE_U16``.
* - __u32 *
  - ``p_u32``
  - A pointer to a matrix control of unsigned 32-bit values. Valid if
    this control is of type ``V4L2_CTRL_TYPE_U32``.
* - struct :c:type:`v4l2_area` *
  - ``p_area``
  - A pointer to a struct :c:type:`v4l2_area`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_AREA``.
* - struct :c:type:`v4l2_ctrl_h264_sps` *
  - ``p_h264_sps``
  - A pointer to a struct :c:type:`v4l2_ctrl_h264_sps`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_H264_SPS``.
* - struct :c:type:`v4l2_ctrl_h264_pps` *
  - ``p_h264_pps``
  - A pointer to a struct :c:type:`v4l2_ctrl_h264_pps`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_H264_PPS``.
* - struct :c:type:`v4l2_ctrl_h264_scaling_matrix` *
  - ``p_h264_scaling_matrix``
  - A pointer to a struct :c:type:`v4l2_ctrl_h264_scaling_matrix`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_H264_SCALING_MATRIX``.
* - struct :c:type:`v4l2_ctrl_h264_pred_weights` *
  - ``p_h264_pred_weights``
  - A pointer to a struct :c:type:`v4l2_ctrl_h264_pred_weights`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_H264_PRED_WEIGHTS``.
* - struct :c:type:`v4l2_ctrl_h264_slice_params` *
  - ``p_h264_slice_params``
  - A pointer to a struct :c:type:`v4l2_ctrl_h264_slice_params`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_H264_SLICE_PARAMS``.
* - struct :c:type:`v4l2_ctrl_h264_decode_params` *
  - ``p_h264_decode_params``
  - A pointer to a struct :c:type:`v4l2_ctrl_h264_decode_params`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_H264_DECODE_PARAMS``.
* - struct :c:type:`v4l2_ctrl_fwht_params` *
  - ``p_fwht_params``
  - A pointer to a struct :c:type:`v4l2_ctrl_fwht_params`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_FWHT_PARAMS``.
* - struct :c:type:`v4l2_ctrl_vp8_frame` *
  - ``p_vp8_frame``
  - A pointer to a struct :c:type:`v4l2_ctrl_vp8_frame`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_VP8_FRAME``.
* - struct :c:type:`v4l2_ctrl_mpeg2_sequence` *
  - ``p_mpeg2_sequence``
  - A pointer to a struct :c:type:`v4l2_ctrl_mpeg2_sequence`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_MPEG2_SEQUENCE``.
* - struct :c:type:`v4l2_ctrl_mpeg2_picture` *
  - ``p_mpeg2_picture``
  - A pointer to a struct :c:type:`v4l2_ctrl_mpeg2_picture`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_MPEG2_PICTURE``.
* - struct :c:type:`v4l2_ctrl_mpeg2_quantisation` *
  - ``p_mpeg2_quantisation``
  - A pointer to a struct :c:type:`v4l2_ctrl_mpeg2_quantisation`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_MPEG2_QUANTISATION``.

```

```

* - struct :c:type:`v4l2_ctrl_vp9_compressed_hdr` *
  - ``p_vp9_compressed_hdr_probs``
  - A pointer to a struct :c:type:`v4l2_ctrl_vp9_compressed_hdr`. Valid if this
    control is of type ``V4L2_CTRL_TYPE_VP9_COMPRESSED_HDR``.
* - struct :c:type:`v4l2_ctrl_vp9_frame` *
  - ``p_vp9_frame``
  - A pointer to a struct :c:type:`v4l2_ctrl_vp9_frame`. Valid if this
    control is of type ``V4L2_CTRL_TYPE_VP9_FRAME``.
* - struct :c:type:`v4l2_ctrl_hdr10_cll_info` *
  - ``p_hdr10_cll``
  - A pointer to a struct :c:type:`v4l2_ctrl_hdr10_cll_info`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_HDR10_CLL_INFO``.
* - struct :c:type:`v4l2_ctrl_hdr10_mastering_display` *
  - ``p_hdr10_mastering``
  - A pointer to a struct :c:type:`v4l2_ctrl_hdr10_mastering_display`. Valid if this control is
    of type ``V4L2_CTRL_TYPE_HDR10_MASTERING_DISPLAY``.
* - void *
  - ``ptr``
  - A pointer to a compound type which can be an N-dimensional array
    and/or a compound type (the control's type is >=
    ``V4L2_CTRL_COMPOUND_TYPES``). Valid if
    ``V4L2_CTRL_FLAG_HAS_PAYLOAD`` is set for this control.
* - }
  -

```

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-g-ext-ctrls.rst, line 265)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{4.0cm}|p{2.5cm}|p{10.8cm}|
```

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-g-ext-ctrls.rst, line 267)

Unknown directive type "c:type".

```
.. c:type:: v4l2_ext_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]vidioc-g-ext-ctrls.rst, line 269)

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-g-ext-ctrls.rst, line 271)

Unknown directive type "flat-table".

```

.. flat-table:: struct v4l2_ext_controls
   :header-rows: 0
   :stub-columns: 0
   :widths:      1 1 2

   * - union {
     - (anonymous)
   * - __u32
     - ``which``
     - Which value of the control to get/set/try.
   * - :cspan: `2` ``V4L2_CTRL_WHICH_CUR_VAL`` will return the current value of
     the control, ``V4L2_CTRL_WHICH_DEF_VAL`` will return the default
     value of the control and ``V4L2_CTRL_WHICH_REQUEST_VAL`` indicates that
     these controls have to be retrieved from a request or tried/set for
     a request. In the latter case the ``request_fd`` field contains the
     file descriptor of the request that should be used. If the device
     does not support requests, then ``EACCES`` will be returned.

```

When using ``V4L2_CTRL_WHICH_DEF_VAL`` be aware that you can only

get the default value of the control, you cannot set or try it.

For backwards compatibility you can also use a control class here (see :ref:`ctrl-class`). In that case all controls have to belong to that control class. This usage is deprecated, instead just use ``V4L2_CTRL_WHICH_CUR_VAL``. There are some very old drivers that do not yet support ``V4L2_CTRL_WHICH_CUR_VAL`` and that require a control class here. You can test for such drivers by setting ``which`` to ``V4L2_CTRL_WHICH_CUR_VAL`` and calling :ref:`VIDIOC_TRY_EXT_CTRL` <VIDIOC_G_EXT_CTRL> with a count of 0. If that fails, then the driver does not support ``V4L2_CTRL_WHICH_CUR_VAL``.

```
* - __u32
- ``ctrl_class``
- Deprecated name kept for backwards compatibility. Use ``which`` instead.
* - }
-
* - __u32
- ``count``
- The number of controls in the controls array. May also be zero.
* - __u32
- ``error_idx``
- Index of the failing control. Set by the driver in case of an error.
* - :cspan:`2` If the error is associated
  with a particular control, then ``error_idx`` is set to the index
  of that control. If the error is not related to a specific
  control, or the validation step failed (see below), then
  ``error_idx`` is set to ``count``. The value is undefined if the
  ioctl returned 0 (success).
```

Before controls are read from/written to hardware a validation step takes place: this checks if all controls in the list are valid controls, if no attempt is made to write to a read-only control or read from a write-only control, and any other up-front checks that can be done without accessing the hardware. The exact validations done during this step are driver dependent since some checks might require hardware access for some devices, thus making it impossible to do those checks up-front. However, drivers should make a best-effort to do as many up-front checks as possible.

This check is done to avoid leaving the hardware in an inconsistent state due to easy-to-avoid problems. But it leads to another problem: the application needs to know whether an error came from the validation step (meaning that the hardware was not touched) or from an error during the actual reading from/writing to hardware.

The, in hindsight quite poor, solution for that is to set ``error_idx`` to ``count`` if the validation failed. This has the unfortunate side-effect that it is not possible to see which control failed the validation. If the validation was successful and the error happened while accessing the hardware, then ``error_idx`` is less than ``count`` and only the controls up to ``error_idx-1`` were read or written correctly, and the state of the remaining controls is undefined.

Since :ref:`VIDIOC_TRY_EXT_CTRL` <VIDIOC_G_EXT_CTRL> does not access hardware there is also no need to handle the validation step in this special way, so ``error_idx`` will just be set to the control that failed the validation step instead of to ``count``. This means that if :ref:`VIDIOC_S_EXT_CTRL` <VIDIOC_G_EXT_CTRL> fails with ``error_idx`` set to ``count``, then you can call :ref:`VIDIOC_TRY_EXT_CTRL` <VIDIOC_G_EXT_CTRL> to try to discover the actual control that failed the validation step. Unfortunately, there is no ``TRY`` equivalent for :ref:`VIDIOC_G_EXT_CTRL` <VIDIOC_G_EXT_CTRL>.

```
* - __s32
- ``request_fd``
- File descriptor of the request to be used by this operation. Only
  valid if ``which`` is set to ``V4L2_CTRL_WHICH_REQUEST_VAL``.
  If the device does not support requests, then ``EACCES`` will be returned.
  If requests are supported but an invalid request file descriptor is
  given, then ``EINVAL`` will be returned.
* - __u32
- ``reserved`` [1]
- Reserved for future extensions.
```

Drivers and applications must set the array to zero.

```
* - struct :c:type:`v4l2_ext_control` *
- ``controls``
- Pointer to an array of ``count`` v4l2_ext_control structures.
```

Ignored if ``count`` equals 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-g-ext-ctrls.rst, line 371)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{7.3cm}|p{2.0cm}|p{8.0cm}|
```

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-g-ext-ctrls.rst, line 373)

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-g-ext-ctrls.rst, line 377)

Unknown directive type "flat-table".

```
.. flat-table:: Control classes
:header-rows: 0
:stub-columns: 0
:widths:      3 1 4

* - ``V4L2_CTRL_CLASS_USER``
  - 0x980000
  - The class containing user controls. These controls are described
    in :ref:`control`. All controls that can be set using the
    :ref:`VIDIOC_S_CTRL` <VIDIOC_G_CTRL>` and
    :ref:`VIDIOC_G_CTRL` <VIDIOC_G_CTRL>` ioctl belong to this
    class.
* - ``V4L2_CTRL_CLASS_CODEC``
  - 0x990000
  - The class containing stateful codec controls. These controls are
    described in :ref:`codec-controls`.
* - ``V4L2_CTRL_CLASS_CAMERA``
  - 0x9a0000
  - The class containing camera controls. These controls are described
    in :ref:`camera-controls`.
* - ``V4L2_CTRL_CLASS_FM_TX``
  - 0x9b0000
  - The class containing FM Transmitter (FM TX) controls. These
    controls are described in :ref:`fm-tx-controls`.
* - ``V4L2_CTRL_CLASS_FLASH``
  - 0x9c0000
  - The class containing flash device controls. These controls are
    described in :ref:`flash-controls`.
* - ``V4L2_CTRL_CLASS_JPEG``
  - 0x9d0000
  - The class containing JPEG compression controls. These controls are
    described in :ref:`jpeg-controls`.
* - ``V4L2_CTRL_CLASS_IMAGE_SOURCE``
  - 0x9e0000
  - The class containing image source controls. These controls are
    described in :ref:`image-source-controls`.
* - ``V4L2_CTRL_CLASS_IMAGE_PROC``
  - 0x9f0000
  - The class containing image processing controls. These controls are
    described in :ref:`image-process-controls`.
* - ``V4L2_CTRL_CLASS_FM_RX``
  - 0xa10000
  - The class containing FM Receiver (FM RX) controls. These controls
    are described in :ref:`fm-rx-controls`.
* - ``V4L2_CTRL_CLASS_RF_TUNER``
  - 0xa20000
  - The class containing RF tuner controls. These controls are
    described in :ref:`rf-tuner-controls`.
* - ``V4L2_CTRL_CLASS_DETECT``
  - 0xa30000
  - The class containing motion or object detection controls. These controls
    are described in :ref:`detect-controls`.
* - ``V4L2_CTRL_CLASS_CODEC_STATELESS``
  - 0xa40000
  - The class containing stateless codec controls. These controls are
    described in :ref:`codec-stateless-controls`.
```



```
* - ``V4L2_CTRL_CLASS_COLORIMETRY``  
- 0xa50000  
- The class containing colorimetry controls. These controls are  
described in :ref:`colorimetry-controls`.
```

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-g-ext-ctrls.rst, line 441); [backlink](#)

Unknown interpreted text role "ref".

EINVAL

The struct `:ctype:`v4l2_ext_control`` `id` is invalid, or the struct `:ctype:`v4l2_ext_controls`` which is invalid, or the struct `:ctype:`v4l2_ext_control`` value was inappropriate (e.g. the given menu index is not supported by the driver), or the `which` field was set to `V4L2_CTRL_WHICH_REQUEST_VAL` but the given `request_fd` was invalid or `V4L2_CTRL_WHICH_REQUEST_VAL` is not supported by the kernel. This error code is also returned by the [ref:VIDIOC_S_EXT_CTRLS <VIDIOC_G_EXT_CTRLS>](#) and [ref:VIDIOC_TRY_EXT_CTRLS <VIDIOC_G_EXT_CTRLS>](#) ioctls if two or more control values are in conflict.

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-g-ext-ctrls.rst, line 446); [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]vidioc-g-ext-ctrls.rst, line 446); [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]vidioc-g-ext-ctrls.rst, line 446); [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]vidioc-g-ext-ctrls.rst, line 446); [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-g-ext-ctrls.rst, line 446); [backlink](#)

Unknown interpreted text role "ref".

ERANGE

The struct `:ctype:`v4l2_ext_control`` value is out of bounds.

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-g-ext-ctrls.rst, line 459); [backlink](#)

Unknown interpreted text role "c.type".

EBUSY

The control is temporarily not changeable, possibly because another applications took over control of the device function

this control belongs to, or (if the `which` field was set to `V4L2_CTRL_WHICH_REQUEST_VAL`) the request was queued but not yet completed.

ENOSPC

The space reserved for the control's payload is insufficient. The field `size` is set to a value that is enough to store the payload and this error code is returned.

EACCES

Attempt to try or set a read-only control, or to get a write-only control, or to get a control from a request that has not yet been completed.

Or the `which` field was set to `V4L2_CTRL_WHICH_REQUEST_VAL` but the device does not support requests.

Or if there is an attempt to set an inactive control and the driver is not capable of caching the new value until the control is active again.