ioctls VIDIOC_QUERYCTRL, VIDIOC_QUERY_EXT_CTRL and VIDIOC_QUERYMENU

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-queryctrl.rst`, line 2)

Unknown directive type "c:namespace".

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

Name

VIDIOC_QUERYCTRL - VIDIOC_QUERY_EXT_CTRL - VIDIOC_QUERYMENU - Enumerate controls and menu control items

Synopsis

```
int ioctl(int fd, int VIDIOC_QUERYCTRL, struct v4l2_queryctrl *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-queryctrl.rst`, line 20)

Unknown directive type "c:macro".

```
.. c:macro:: VIDIOC_QUERY_EXT_CTRL
```

```
int ioctl(int fd, VIDIOC_QUERY_EXT_CTRL, struct v4l2_query_ext_ctrl *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-queryctrl.rst`, line 24)

Unknown directive type "c:macro".

```
.. c:macro:: VIDIOC_QUERYMENU
```

```
int ioctl(int fd, VIDIOC_QUERYMENU, struct v4l2_querymenu *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-queryctrl.rst`, line 32); *backlink*

Unknown interpreted text role "c:func".

`argp`

Pointer to struct :c:type:`v4l2_queryctrl`, :c:type:`v4l2_query_ext_ctrl` or :c:type:`v4l2_querymenu` (depending on the ioctl).

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-queryctrl.rst`, line 35); *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-queryctrl.rst`, line 35); *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-queryctrl.rst`, line 35); *backlink*

Unknown interpreted text role "c:type".

Description

To query the attributes of a control applications set the `id` field of a struct :ref:`v4l2_queryctrl <v4l2-queryctrl>` and call the `VIDIOC_QUERYCTRL` ioctl with a pointer to this structure. The driver fills the rest of the structure or returns an `EINVAL` error code when the `id` is invalid.

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-queryctrl.rst`, line 41); *backlink*

Unknown interpreted text role "ref".

It is possible to enumerate controls by calling `VIDIOC_QUERYCTRL` with successive `id` values starting from `V4L2_CID_BASE` up to and exclusive `V4L2_CID_LASTP1`. Drivers may return `EINVAL` if a control in this range is not supported. Further applications can enumerate private controls, which are not defined in this specification, by starting at `V4L2_CID_PRIVATE_BASE` and incrementing `id` until the driver returns `EINVAL`.

In both cases, when the driver sets the `V4L2_CTRL_FLAG_DISABLED` flag in the `flags` field this control is permanently disabled and should be ignored by the application. [1]

When the application ORs `id` with `V4L2_CTRL_FLAG_NEXT_CTRL` the driver returns the next supported non-compound control, or `EINVAL` if there is none. In addition, the `V4L2_CTRL_FLAG_NEXT_COMPOUND` flag can be specified to enumerate all compound controls (i.e. controls with type â‰¥ `V4L2_CTRL_COMPOUND_TYPES` and/or array control, in other words controls that contain more than one value). Specify both `V4L2_CTRL_FLAG_NEXT_CTRL` and `V4L2_CTRL_FLAG_NEXT_COMPOUND` in order to enumerate all controls, compound or not. Drivers which do not support these flags yet always return `EINVAL`.

The `VIDIOC_QUERY_EXT_CTRL` ioctl was introduced in order to better support controls that can use compound types, and to expose additional control information that cannot be returned in struct :ref:`v4l2_queryctrl <v4l2-queryctrl>` since that structure is full.

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-queryctrl.rst`, line 69); *backlink*

Unknown interpreted text role "ref".

`VIDIOC_QUERY_EXT_CTRL` is used in the same way as `VIDIOC_QUERYCTRL`, except that the `reserved` array must be zeroed as well.

Additional information is required for menu controls: the names of the menu items. To query them applications set the `id` and `index` fields of struct :ref:`v4l2_querymenu <v4l2-querymenu>` and call the `VIDIOC_QUERYMENU` ioctl with a pointer to this structure. The driver fills the rest of the structure or returns an `EINVAL` error code when the `id` or `index` is invalid. Menu items are enumerated by calling `VIDIOC_QUERYMENU` with successive `index` values from struct :ref:`v4l2_queryctrl <v4l2-queryctrl>` `minimum` to `maximum`, inclusive.

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-queryctrl.rst`, line 78); *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-queryctrl.rst`, line 78); *backlink*

Unknown interpreted text role "ref".

Note

It is possible for `VIDIOC_QUERYMENU` to return an `EINVAL` error code for some indices between `minimum` and `maximum`. In that case that particular menu item is not supported by this driver. Also note that the `minimum` value is not necessarily 0.

See also the examples in :ref:`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-queryctrl.rst`, line 95); *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-queryctrl.rst`, line 97)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{1.2cm}|p{3.6cm}|p{12.5cm}|
```

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-queryctrl.rst`, line 101)

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-queryctrl.rst`, line 103)

Unknown directive type "flat-table".

```
.. flat-table:: struct v4l2_queryctrl
    :header-rows:  0
    :stub-columns: 0
    :widths:       1 1 2

    * - __u32
      - ``id``
      - Identifies the control, set by the application. See
        :ref:`control-id` for predefined IDs. When the ID is ORed with
        V4L2_CTRL_FLAG_NEXT_CTRL the driver clears the flag and
        returns the first control with a higher ID. Drivers which do not
        support this flag yet always return an ``EINVAL`` error code.
    * - __u32
      - ``type``
      - Type of control, see :c:type:`v4l2_ctrl_type`.
    * - __u8
      - ``name``\ [32]
      - Name of the control, a NUL-terminated ASCII string. This
        information is intended for the user.
    * - __s32
      - ``minimum``
      - Minimum value, inclusive. This field gives a lower bound for the
        control. See enum :c:type:`v4l2_ctrl_type` how
        the minimum value is to be used for each possible control type.
        Note that this a signed 32-bit value.
    * - __s32
      - ``maximum``
      - Maximum value, inclusive. This field gives an upper bound for the
        control. See enum :c:type:`v4l2_ctrl_type` how
        the maximum value is to be used for each possible control type.
        Note that this a signed 32-bit value.
    * - __s32
      - ``step``
      - This field gives a step size for the control. See enum
```

```
          :c:type:`v4l2_ctrl_type` how the step value is
          to be used for each possible control type. Note that this an
          unsigned 32-bit value.

          Generally drivers should not scale hardware control values. It may
          be necessary for example when the ``name`` or ``id`` imply a
          particular unit and the hardware actually accepts only multiples
          of said unit. If so, drivers must take care values are properly
          rounded when scaling, such that errors will not accumulate on
          repeated read-write cycles.

          This field gives the smallest change of an integer control
          actually affecting hardware. Often the information is needed when
          the user can change controls by keyboard or GUI buttons, rather
          than a slider. When for example a hardware register accepts values
          0-511 and the driver reports 0-65535, step should be 128.

          Note that although signed, the step value is supposed to be always
          positive.
      * - __s32
        - ``default_value``
        - The default value of a ``V4L2_CTRL_TYPE_INTEGER``, ``_BOOLEAN``,
          ``_BITMASK``, ``_MENU`` or ``_INTEGER_MENU`` control. Not valid
          for other types of controls.

          .. note::

             Drivers reset controls to their default value only when
             the driver is first loaded, never afterwards.
      * - __u32
        - ``flags``
        - Control flags, see :ref:`control-flags`.
      * - __u32
        - ``reserved``\ [2]
        - Reserved for future extensions. Drivers 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-queryctrl.rst`, **line 175**)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{1.2cm}|p{5.5cm}|p{10.6cm}|
```

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-queryctrl.rst`, **line 179**)

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-queryctrl.rst`, **line 181**)

Unknown directive type "flat-table".

```
.. flat-table:: struct v4l2_query_ext_ctrl
    :header-rows:  0
    :stub-columns: 0
    :widths:       1 1 2

    * - __u32
      - ``id``
      - Identifies the control, set by the application. See
        :ref:`control-id` for predefined IDs. When the ID is ORed with
        ``V4L2_CTRL_FLAG_NEXT_CTRL`` the driver clears the flag and
        returns the first non-compound control with a higher ID. When the
        ID is ORed with ``V4L2_CTRL_FLAG_NEXT_COMPOUND`` the driver clears
        the flag and returns the first compound control with a higher ID.
        Set both to get the first control (compound or not) with a higher
        ID.
    * - __u32
      - ``type``
      - Type of control, see :c:type:`v4l2_ctrl_type`.
    * - char
      - ``name``\ [32]
      - Name of the control, a NUL-terminated ASCII string. This
        information is intended for the user.
    * - __s64
      - ``minimum``
      - Minimum value, inclusive. This field gives a lower bound for the
        control. See enum :c:type:`v4l2_ctrl_type` how
        the minimum value is to be used for each possible control type.
        Note that this a signed 64-bit value.
    * - __s64
      - ``maximum``
      - Maximum value, inclusive. This field gives an upper bound for the
        control. See enum :c:type:`v4l2_ctrl_type` how
        the maximum value is to be used for each possible control type.
        Note that this a signed 64-bit value.
    * - __u64
      - ``step``
      - This field gives a step size for the control. See enum
        :c:type:`v4l2_ctrl_type` how the step value is
        to be used for each possible control type. Note that this an
        unsigned 64-bit value.

        Generally drivers should not scale hardware control values. It may
        be necessary for example when the ``name`` or ``id`` imply a
        particular unit and the hardware actually accepts only multiples
        of said unit. If so, drivers must take care values are properly
        rounded when scaling, such that errors will not accumulate on
        repeated read-write cycles.

        This field gives the smallest change of an integer control
        actually affecting hardware. Often the information is needed when
        the user can change controls by keyboard or GUI buttons, rather
        than a slider. When for example a hardware register accepts values
        0-511 and the driver reports 0-65535, step should be 128.
    * - __s64
      - ``default_value``
      - The default value of a ``V4L2_CTRL_TYPE_INTEGER``, ``_INTEGER64``,
        ``_BOOLEAN``, ``_BITMASK``, ``_MENU``, ``_INTEGER_MENU``, ``_U8``
        or ``_U16`` control. Not valid for other types of controls.

        .. note::

           Drivers reset controls to their default value only when
           the driver is first loaded, never afterwards.
    * - __u32
      - ``flags``
      - Control flags, see :ref:`control-flags`.
    * - __u32
      - ``elem_size``
      - The size in bytes of a single element of the array. Given a char
        pointer ``p`` to a 3-dimensional array you can find the position
        of cell ``(z, y, x)`` as follows:
        ``p + ((z * dims[1] + y) * dims[0] + x) * elem_size``.
        ``elem_size`` is always valid, also when the control isn't an
        array. For string controls ``elem_size`` is equal to
        ``maximum + 1``.
    * - __u32
      - ``elems``
      - The number of elements in the N-dimensional array. If this control
        is not an array, then ``elems`` is 1. The ``elems`` field can
        never be 0.
    * - __u32
      - ``nr_of_dims``
      - The number of dimension in the N-dimensional array. If this
        control is not an array, then this field is 0.
    * - __u32
      - ``dims[V4L2_CTRL_MAX_DIMS]``
      - The size of each dimension. The first ``nr_of_dims`` elements of
        this array must be non-zero, all remaining elements must be zero.
    * - __u32
      - ``reserved``\ [32]
      - Reserved for future extensions. Applications and drivers 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-queryctrl.rst`, line 275)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{1.2cm}|p{3.0cm}|p{13.1cm}|
```

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-queryctrl.rst`, line 279)

Unknown directive type "flat-table".

```
.. flat-table:: struct v4l2_querymenu
    :header-rows:  0
    :stub-columns: 0
    :widths:       1 1 2

    * - __u32
      - ``id``
      - Identifies the control, set by the application from the respective
        struct :ref:`v4l2_queryctrl <v4l2-queryctrl>` ``id``.
    * - __u32
      - ``index``
      - Index of the menu item, starting at zero, set by the application.
    * - union {
      - (anonymous)
    * - __u8
      - ``name``\ [32]
      - Name of the menu item, a NUL-terminated ASCII string. This
        information is intended for the user. This field is valid for
        ``V4L2_CTRL_TYPE_MENU`` type controls.
    * - __s64
      - ``value``
      - Value of the integer menu item. This field is valid for
        ``V4L2_CTRL_TYPE_INTEGER_MENU`` type controls.
    * - }
      -
    * - __u32
      - ``reserved``
      - Reserved for future extensions. Drivers 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-queryctrl.rst`, line 309)

Unknown directive type "c:type".

```
.. c:type:: v4l2_ctrl_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-queryctrl.rst`, line 315)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{6.5cm}|p{1.5cm}|p{1.1cm}|p{1.5cm}|p{6.8cm}|
```

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-queryctrl.rst`, line 317)

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-queryctrl.rst, line 319)

Unknown directive type "flat-table".

```
.. flat-table:: enum v4l2_ctrl_type
    :header-rows:  1
    :stub-columns: 0
    :widths:       30 5 5 5 55

    * - Type
      - ``minimum``
      - ``step``
      - ``maximum``
      - Description
    * - ``V4L2_CTRL_TYPE_INTEGER``
      - any
      - any
      - any
      - An integer-valued control ranging from minimum to maximum
        inclusive. The step value indicates the increment between values.
    * - ``V4L2_CTRL_TYPE_BOOLEAN``
      - 0
      - 1
      - 1
      - A boolean-valued control. Zero corresponds to "disabled", and one
        means "enabled".
    * - ``V4L2_CTRL_TYPE_MENU``
      - â‰¥ 0
      - 1
      - N-1
      - The control has a menu of N choices. The names of the menu items
        can be enumerated with the ``VIDIOC_QUERYMENU`` ioctl.
    * - ``V4L2_CTRL_TYPE_INTEGER_MENU``
      - â‰¥ 0
      - 1
      - N-1
      - The control has a menu of N choices. The values of the menu items
        can be enumerated with the ``VIDIOC_QUERYMENU`` ioctl. This is
        similar to ``V4L2_CTRL_TYPE_MENU`` except that instead of strings,
        the menu items are signed 64-bit integers.
    * - ``V4L2_CTRL_TYPE_BITMASK``
      - 0
      - n/a
      - any
      - A bitmask field. The maximum value is the set of bits that can be
        used, all other bits are to be 0. The maximum value is interpreted
        as a __u32, allowing the use of bit 31 in the bitmask.
    * - ``V4L2_CTRL_TYPE_BUTTON``
      - 0
      - 0
      - 0
      - A control which performs an action when set. Drivers must ignore
        the value passed with ``VIDIOC_S_CTRL`` and return an ``EACCES`` error
        code on a ``VIDIOC_G_CTRL`` attempt.
    * - ``V4L2_CTRL_TYPE_INTEGER64``
      - any
      - any
      - any
      - A 64-bit integer valued control. Minimum, maximum and step size
        cannot be queried using ``VIDIOC_QUERYCTRL``. Only
        ``VIDIOC_QUERY_EXT_CTRL`` can retrieve the 64-bit min/max/step
        values, they should be interpreted as n/a when using
        ``VIDIOC_QUERYCTRL``.
    * - ``V4L2_CTRL_TYPE_STRING``
      - â‰¥ 0
      - â‰¥ 1
      - â‰¥ 0
      - The minimum and maximum string lengths. The step size means that
        the string must be (minimum + N * step) characters long for N â‰¥ 0.
        These lengths do not include the terminating zero, so in order to
        pass a string of length 8 to
        :ref:`VIDIOC_S_EXT_CTRLS <VIDIOC_G_EXT_CTRLS>` you need to
        set the ``size`` field of struct
        :c:type:`v4l2_ext_control` to 9. For
        :ref:`VIDIOC_G_EXT_CTRLS <VIDIOC_G_EXT_CTRLS>` you can set
        the ``size`` field to ``maximum`` + 1. Which character encoding is
        used will depend on the string control itself and should be part
        of the control documentation.
    * - ``V4L2_CTRL_TYPE_CTRL_CLASS``
      - n/a
      - n/a
      - n/a
      - This is not a control. When ``VIDIOC_QUERYCTRL`` is called with a
        control ID equal to a control class code (see :ref:`ctrl-class`)
        + 1, the ioctl returns the name of the control class and this
        control type. Older drivers which do not support this feature
        return an ``EINVAL`` error code.
    * - ``V4L2_CTRL_TYPE_U8``
      - any
      - any
      - any
      - An unsigned 8-bit valued control ranging from minimum to maximum
        inclusive. The step value indicates the increment between values.
    * - ``V4L2_CTRL_TYPE_U16``
      - any
      - any
      - any
      - An unsigned 16-bit valued control ranging from minimum to maximum
        inclusive. The step value indicates the increment between values.
    * - ``V4L2_CTRL_TYPE_U32``
      - any
      - any
      - any
      - An unsigned 32-bit valued control ranging from minimum to maximum
        inclusive. The step value indicates the increment between values.
    * - ``V4L2_CTRL_TYPE_MPEG2_QUANTISATION``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_mpeg2_quantisation`, containing MPEG-2
        quantisation matrices for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_MPEG2_SEQUENCE``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_mpeg2_sequence`, containing MPEG-2
        sequence parameters for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_MPEG2_PICTURE``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_mpeg2_picture`, containing MPEG-2
        picture parameters for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_AREA``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_area`, containing the width and the height
        of a rectangular area. Units depend on the use case.
    * - ``V4L2_CTRL_TYPE_H264_SPS``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_h264_sps`, containing H264
        sequence parameters for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_H264_PPS``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_h264_pps`, containing H264
        picture parameters for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_H264_SCALING_MATRIX``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_h264_scaling_matrix`, containing H264
        scaling matrices for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_H264_SLICE_PARAMS``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_h264_slice_params`, containing H264
        slice parameters for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_H264_DECODE_PARAMS``
      - n/a
      - n/a
      - n/a
      - A struct :c:type:`v4l2_ctrl_h264_decode_params`, containing H264
        decode parameters for stateless video decoders.
    * - ``V4L2_CTRL_TYPE_FWHT_PARAMS``
      - n/a
```

```
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_fwht_params`, containing FWHT
            parameters for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_HEVC_SPS``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_hevc_sps`, containing HEVC Sequence
            Parameter Set for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_HEVC_PPS``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_hevc_pps`, containing HEVC Picture
            Parameter Set for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_HEVC_SLICE_PARAMS``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_hevc_slice_params`, containing HEVC
            slice parameters for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_HEVC_SCALING_MATRIX``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_hevc_scaling_matrix`, containing HEVC
            scaling matrix for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_VP8_FRAME``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_vp8_frame`, containing VP8
            frame parameters for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_HEVC_DECODE_PARAMS``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_hevc_decode_params`, containing HEVC
            decoding parameters for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_VP9_COMPRESSED_HDR``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_vp9_compressed_hdr`, containing VP9
            probabilities updates for stateless video decoders.
        * - ``V4L2_CTRL_TYPE_VP9_FRAME``
          - n/a
          - n/a
          - n/a
          - A struct :c:type:`v4l2_ctrl_vp9_frame`, containing VP9
            frame decode parameters for stateless video 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-queryctrl.rst`, line 533)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{7.3cm}|p{1.8cm}|p{8.2cm}|
```

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-queryctrl.rst`, line 535)

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-queryctrl.rst`, line 539)

Unknown directive type "flat-table".

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

    * - ``V4L2_CTRL_FLAG_DISABLED``
      - 0x0001
      - This control is permanently disabled and should be ignored by the
        application. Any attempt to change the control will result in an
        ``EINVAL`` error code.
    * - ``V4L2_CTRL_FLAG_GRABBED``
      - 0x0002
      - This control is temporarily unchangeable, for example because
        another application took over control of the respective resource.
        Such controls may be displayed specially in a user interface.
        Attempts to change the control may result in an ``EBUSY`` error code.
    * - ``V4L2_CTRL_FLAG_READ_ONLY``
      - 0x0004
      - This control is permanently readable only. Any attempt to change
        the control will result in an ``EINVAL`` error code.
    * - ``V4L2_CTRL_FLAG_UPDATE``
      - 0x0008
      - A hint that changing this control may affect the value of other
        controls within the same control class. Applications should update
        their user interface accordingly.
    * - ``V4L2_CTRL_FLAG_INACTIVE``
      - 0x0010
      - This control is not applicable to the current configuration and
        should be displayed accordingly in a user interface. For example
        the flag may be set on a MPEG audio level 2 bitrate control when
        MPEG audio encoding level 1 was selected with another control.
    * - ``V4L2_CTRL_FLAG_SLIDER``
      - 0x0020
      - A hint that this control is best represented as a slider-like
        element in a user interface.
    * - ``V4L2_CTRL_FLAG_WRITE_ONLY``
      - 0x0040
      - This control is permanently writable only. Any attempt to read the
        control will result in an ``EACCES`` error code error code. This flag
        is typically present for relative controls or action controls
        where writing a value will cause the device to carry out a given
        action (e. g. motor control) but no meaningful value can be
        returned.
    * - ``V4L2_CTRL_FLAG_VOLATILE``
      - 0x0080
      - This control is volatile, which means that the value of the
        control changes continuously. A typical example would be the
        current gain value if the device is in auto-gain mode. In such a
        case the hardware calculates the gain value based on the lighting
        conditions which can change over time.

        .. note::

           Setting a new value for a volatile control will be ignored
           unless
           :ref:`V4L2_CTRL_FLAG_EXECUTE_ON_WRITE <FLAG_EXECUTE_ON_WRITE>`
           is also set.
           Setting a new value for a volatile control will *never* trigger a
           :ref:`V4L2_EVENT_CTRL_CH_VALUE <ctrl-changes-flags>` event.
    * - ``V4L2_CTRL_FLAG_HAS_PAYLOAD``
      - 0x0100
      - This control has a pointer type, so its value has to be accessed
        using one of the pointer fields of struct
        :c:type:`v4l2_ext_control`. This flag is set
        for controls that are an array, string, or have a compound type.
        In all cases you have to set a pointer to memory containing the
        payload of the control.
    * .. _FLAG_EXECUTE_ON_WRITE:

      - ``V4L2_CTRL_FLAG_EXECUTE_ON_WRITE``
      - 0x0200
      - The value provided to the control will be propagated to the driver
        even if it remains constant. This is required when the control
        represents an action on the hardware. For example: clearing an
        error flag or triggering the flash. All the controls of the type
        ``V4L2_CTRL_TYPE_BUTTON`` have this flag set.
    * .. _FLAG_MODIFY_LAYOUT:

      - ``V4L2_CTRL_FLAG_MODIFY_LAYOUT``
      - 0x0400
      - Changing this control value may modify the layout of the
```

```
buffer (for video devices) or the media bus format (for sub-devices).

A typical example would be the ``V4L2_CID_ROTATE`` control.

Note that typically controls with this flag will also set the
``V4L2_CTRL_FLAG_GRABBED`` flag when buffers are allocated or
streaming is in progress since most drivers do not support changing
the format in that case.
```

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-queryctrl.rst`, line 632); ***backlink***

Unknown interpreted text role "ref".

EINVAL

The struct :ref:`v4l2_queryctrl <v4l2-queryctrl>` `id` is invalid. The struct :ref:`v4l2_querymenu <v4l2-querymenu>` `id` is invalid or `index` is out of range (less than `minimum` or greater than `maximum`) or this particular menu item is not supported by the driver.

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-queryctrl.rst`, line 637); ***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-queryctrl.rst`, line 637); ***backlink***

Unknown interpreted text role "ref".

EACCES

An attempt was made to read a write-only control.

[1] `V4L2_CTRL_FLAG_DISABLED` was intended for two purposes: Drivers can skip predefined controls not supported by the hardware (although returning `EINVAL` would do as well), or disable predefined and private controls after hardware detection without the trouble of reordering control arrays and indices (`EINVAL` cannot be used to skip private controls because it would prematurely end the enumeration).