ioctl VIDIOC_QUERYCAP

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-querycap.rst, line 2)

Unknown directive type "c:namespace".

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

Name

VIDIOC_QUERYCAP - Query device capabilities

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-querycap.rst, line 18)

Unknown directive type "c:macro".

```
.. c:macro:: VIDIOC_QUERYCAP
```

```
int ioctl(int fd, VIDIOC_QUERYCAP, struct v4l2_capability *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-querycap.rst, line 26); *backlink*

Unknown interpreted text role "c:func".

`argp`

Pointer to struct :c:type:`v4l2_capability`.

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-querycap.rst, line 29); *backlink*

Unknown interpreted text role "c:type".

Description

All V4L2 devices support the `VIDIOC_QUERYCAP` ioctl. It is used to identify kernel devices compatible with this specification and to obtain information about driver and hardware capabilities. The ioctl takes a pointer to a struct :c:type:`v4l2_capability` which is filled by the driver. When the driver is not compatible with this specification the ioctl returns an `EINVAL` error code.

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-querycap.rst, line 34); *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-querycap.rst, line 41)

Unknown directive type "c:type".

```
.. c:type:: v4l2_capability
```

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-querycap.rst, line 43)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{1.4cm}|p{2.8cm}|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-querycap.rst, line 45)

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-querycap.rst, line 47)

Unknown directive type "flat-table".

```
.. flat-table:: struct v4l2_capability
    :header-rows:  0
    :stub-columns: 0
    :widths:       3 4 20

    * - __u8
      - ``driver``\ [16]
      - Name of the driver, a unique NUL-terminated ASCII string. For
        example: "bttv". Driver specific applications can use this
        information to verify the driver identity. It is also useful to
        work around known bugs, or to identify drivers in error reports.

        Storing strings in fixed sized arrays is bad practice but
        unavoidable here. Drivers and applications should take precautions
        to never read or write beyond the end of the array and to make
        sure the strings are properly NUL-terminated.
    * - __u8
      - ``card``\ [32]
      - Name of the device, a NUL-terminated UTF-8 string. For example:
        "Yoyodyne TV/FM". One driver may support different brands or
        models of video hardware. This information is intended for users,
        for example in a menu of available devices. Since multiple TV
        cards of the same brand may be installed which are supported by
        the same driver, this name should be combined with the character
        device file name (e. g. ``/dev/video2``) or the ``bus_info``
        string to avoid ambiguities.
    * - __u8
      - ``bus_info``\ [32]
      - Location of the device in the system, a NUL-terminated ASCII
        string. For example: "PCI:0000:05:06.0". This information is
        intended for users, to distinguish multiple identical devices. If
        no such information is available the field must simply count the
        devices controlled by the driver ("platform:vivid-000"). The
        bus_info must start with "PCI:" for PCI boards, "PCIe:" for PCI
        Express boards, "usb-" for USB devices, "I2C:" for i2c devices,
        "ISA:" for ISA devices, "parport" for parallel port devices and
        "platform:" for platform devices.
    * - __u32
      - ``version``
      - Version number of the driver.

        Starting with kernel 3.1, the version reported is provided by the
        V4L2 subsystem following the kernel numbering scheme. However, it
        may not always return the same version as the kernel if, for
        example, a stable or distribution-modified kernel uses the V4L2
        stack from a newer kernel.

        The version number is formatted using the ``KERNEL_VERSION()``
```

```
      macro. For example if the media stack corresponds to the V4L2
      version shipped with Kernel 4.14, it would be equivalent to:
    * - :cspan:`2`

      ``#define KERNEL_VERSION(a,b,c) (((a) << 16) + ((b) << 8) + (c))``

      ``__u32 version = KERNEL_VERSION(4, 14, 0);``

      ``printf ("Version: %u.%u.%u\\n",``

      ``(version >> 16) & 0xFF, (version >> 8) & 0xFF, version & 0xFF);``
    * - __u32
      - ``capabilities``
      - Available capabilities of the physical device as a whole, see
        :ref:`device-capabilities`. The same physical device can export
        multiple devices in /dev (e.g. /dev/videoX, /dev/vbiY and
        /dev/radioZ). The ``capabilities`` field should contain a union of
        all capabilities available around the several V4L2 devices
        exported to userspace. For all those devices the ``capabilities``
        field returns the same set of capabilities. This allows
        applications to open just one of the devices (typically the video
        device) and discover whether video, vbi and/or radio are also
        supported.
    * - __u32
      - ``device_caps``
      - Device capabilities of the opened device, see
        :ref:`device-capabilities`. Should contain the available
        capabilities of that specific device node. So, for example,
        ``device_caps`` of a radio device will only contain radio related
        capabilities and no video or vbi capabilities. This field is only
        set if the ``capabilities`` field contains the
        ``V4L2_CAP_DEVICE_CAPS`` capability. Only the ``capabilities``
        field can have the ``V4L2_CAP_DEVICE_CAPS`` capability,
        ``device_caps`` will never set ``V4L2_CAP_DEVICE_CAPS``.
    * - __u32
      - ``reserved``\ [3]
      - Reserved for future extensions. Drivers must set this 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-querycap.rst, line 135)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{7.0cm}|p{2.6cm}|p{7.7cm}|
```

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-querycap.rst, line 139)

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-querycap.rst, line 141)

Unknown directive type "flat-table".

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

    * - ``V4L2_CAP_VIDEO_CAPTURE``
      - 0x00000001
      - The device supports the single-planar API through the
        :ref:`Video Capture <capture>` interface.
    * - ``V4L2_CAP_VIDEO_CAPTURE_MPLANE``
      - 0x00001000
      - The device supports the :ref:`multi-planar API <planar-apis>`
        through the :ref:`Video Capture <capture>` interface.
```

* - ``V4L2_CAP_VIDEO_OUTPUT``
  - 0x00000002
  - The device supports the single-planar API through the :ref:`Video Output <output>` interface.
* - ``V4L2_CAP_VIDEO_OUTPUT_MPLANE``
  - 0x00002000
  - The device supports the :ref:`multi-planar API <planar-apis>` through the :ref:`Video Output <output>` interface.
* - ``V4L2_CAP_VIDEO_M2M``
  - 0x00008000
  - The device supports the single-planar API through the Video Memory-To-Memory interface.
* - ``V4L2_CAP_VIDEO_M2M_MPLANE``
  - 0x00004000
  - The device supports the :ref:`multi-planar API <planar-apis>` through the Video Memory-To-Memory interface.
* - ``V4L2_CAP_VIDEO_OVERLAY``
  - 0x00000004
  - The device supports the :ref:`Video Overlay <overlay>` interface. A video overlay device typically stores captured images directly in the video memory of a graphics card, with hardware clipping and scaling.
* - ``V4L2_CAP_VBI_CAPTURE``
  - 0x00000010
  - The device supports the :ref:`Raw VBI Capture <raw-vbi>` interface, providing Teletext and Closed Caption data.
* - ``V4L2_CAP_VBI_OUTPUT``
  - 0x00000020
  - The device supports the :ref:`Raw VBI Output <raw-vbi>` interface.
* - ``V4L2_CAP_SLICED_VBI_CAPTURE``
  - 0x00000040
  - The device supports the :ref:`Sliced VBI Capture <sliced>` interface.
* - ``V4L2_CAP_SLICED_VBI_OUTPUT``
  - 0x00000080
  - The device supports the :ref:`Sliced VBI Output <sliced>` interface.
* - ``V4L2_CAP_RDS_CAPTURE``
  - 0x00000100
  - The device supports the :ref:`RDS <rds>` capture interface.
* - ``V4L2_CAP_VIDEO_OUTPUT_OVERLAY``
  - 0x00000200
  - The device supports the :ref:`Video Output Overlay <osd>` (OSD) interface. Unlike the *Video Overlay* interface, this is a secondary function of video output devices and overlays an image onto an outgoing video signal. When the driver sets this flag, it must clear the ``V4L2_CAP_VIDEO_OVERLAY`` flag and vice versa. [#f1]_
* - ``V4L2_CAP_HW_FREQ_SEEK``
  - 0x00000400
  - The device supports the :ref:`VIDIOC_S_HW_FREQ_SEEK` ioctl for hardware frequency seeking.
* - ``V4L2_CAP_RDS_OUTPUT``
  - 0x00000800
  - The device supports the :ref:`RDS <rds>` output interface.
* - ``V4L2_CAP_TUNER``
  - 0x00010000
  - The device has some sort of tuner to receive RF-modulated video signals. For more information about tuner programming see :ref:`tuner`.
* - ``V4L2_CAP_AUDIO``
  - 0x00020000
  - The device has audio inputs or outputs. It may or may not support audio recording or playback, in PCM or compressed formats. PCM audio support must be implemented as ALSA or OSS interface. For more information on audio inputs and outputs see :ref:`audio`.
* - ``V4L2_CAP_RADIO``
  - 0x00040000
  - This is a radio receiver.
* - ``V4L2_CAP_MODULATOR``
  - 0x00080000
  - The device has some sort of modulator to emit RF-modulated video/audio signals. For more information about modulator programming see :ref:`tuner`.
* - ``V4L2_CAP_SDR_CAPTURE``
  - 0x00100000
  - The device supports the :ref:`SDR Capture <sdr>` interface.
* - ``V4L2_CAP_EXT_PIX_FORMAT``
  - 0x00200000

```
      - The device supports the struct
        :c:type:`v4l2_pix_format` extended fields.
    * - ``V4L2_CAP_SDR_OUTPUT``
      - 0x00400000
      - The device supports the :ref:`SDR Output <sdr>` interface.
    * - ``V4L2_CAP_META_CAPTURE``
      - 0x00800000
      - The device supports the :ref:`metadata` capture interface.
    * - ``V4L2_CAP_READWRITE``
      - 0x01000000
      - The device supports the :c:func:`read()` and/or
        :c:func:`write()` I/O methods.
    * - ``V4L2_CAP_ASYNCIO``
      - 0x02000000
      - The device supports the :ref:`asynchronous <async>` I/O methods.
    * - ``V4L2_CAP_STREAMING``
      - 0x04000000
      - The device supports the :ref:`streaming <mmap>` I/O method.
    * - ``V4L2_CAP_META_OUTPUT``
      - 0x08000000
      - The device supports the :ref:`metadata` output interface.
    * - ``V4L2_CAP_TOUCH``
      - 0x10000000
      - This is a touch device.
    * - ``V4L2_CAP_IO_MC``
      - 0x20000000
      - There is only one input and/or output seen from userspace. The whole
        video topology configuration, including which I/O entity is routed to
        the input/output, is configured by userspace via the Media Controller.
        See :ref:`media_controller`.
    * - ``V4L2_CAP_DEVICE_CAPS``
      - 0x80000000
      - The driver fills the ``device_caps`` field. This capability can
        only appear in the ``capabilities`` field and never in the
        ``device_caps`` field.
```

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-querycap.rst, line 274); *backlink*

Unknown interpreted text role "ref".

[1] The struct :c:type:`v4l2_framebuffer` lacks an enum :c:type:`v4l2_buf_type` field, therefore the type of overlay is implied by the driver 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]vidioc-querycap.rst, line 279); *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-querycap.rst, line 279); *backlink*

Unknown interpreted text role "c:type".