ioctl VIDIOC_SUBDEV_G_FMT, VIDIOC_SUBDEV_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]vidioc-subdev-g-fmt.rst, line 2`)**
>
> Unknown directive type "c:namespace".
>
> ```
>     .. c:namespace:: V4L
> ```

Name

VIDIOC_SUBDEV_G_FMT - VIDIOC_SUBDEV_S_FMT - Get or set the data format on a subdev pad

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-subdev-g-fmt.rst, line 18`)**
>
> Unknown directive type "c:macro".
>
> ```
>     .. c:macro:: VIDIOC_SUBDEV_G_FMT
> ```

```
int ioctl(int fd, VIDIOC_SUBDEV_G_FMT, struct v4l2_subdev_format *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-subdev-g-fmt.rst, line 22`)**
>
> Unknown directive type "c:macro".
>
> ```
>     .. c:macro:: VIDIOC_SUBDEV_S_FMT
> ```

```
int ioctl(int fd, VIDIOC_SUBDEV_S_FMT, struct v4l2_subdev_format *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-subdev-g-fmt.rst, line 30`); *backlink***
>
> Unknown interpreted text role "c:func".

`argp`

Pointer to struct :c:type:`v4l2_subdev_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]vidioc-subdev-g-fmt.rst, line 33`); *backlink***
>
> Unknown interpreted text role "c:type".

Description

These ioctls are used to negotiate the frame format at specific subdev pads in the image pipeline.

To retrieve the current format applications set the `pad` field of a struct :c:type:`v4l2_subdev_format` to the desired pad number as reported by the media API and the `which` field to `V4L2_SUBDEV_FORMAT_ACTIVE`. When they call the `VIDIOC_SUBDEV_G_FMT` ioctl with a pointer to this structure the driver fills the members of the `format` field.

To change the current format applications set both the `pad` and `which` fields and all members of the `format` field. When they call the `VIDIOC_SUBDEV_S_FMT` ioctl with a pointer to this structure the driver verifies the requested format, adjusts it based on the hardware capabilities and configures the device. Upon return the struct :c:type:`v4l2_subdev_format` contains the current format as would be returned by a `VIDIOC_SUBDEV_G_FMT` call.

Applications can query the device capabilities by setting the `which` to `V4L2_SUBDEV_FORMAT_TRY`. When set, 'try' formats are not applied to the device by the driver, but are changed exactly as active formats and stored in the sub-device file handle. Two applications querying the same sub-device would thus not interact with each other.

For instance, to try a format at the output pad of a sub-device, applications would first set the try format at the sub-device input with the `VIDIOC_SUBDEV_S_FMT` ioctl. They would then either retrieve the default format at the output pad with the `VIDIOC_SUBDEV_G_FMT` ioctl, or set the desired output pad format with the `VIDIOC_SUBDEV_S_FMT` ioctl and check the returned value.

Try formats do not depend on active formats, but can depend on the current links configuration or sub-device controls value. For instance, a low-pass noise filter might crop pixels at the frame boundaries, modifying its output frame size.

If the subdev device node has been registered in read-only mode, calls to `VIDIOC_SUBDEV_S_FMT` are only valid if the `which` field is set to `V4L2_SUBDEV_FORMAT_TRY`, otherwise an error is returned and the errno variable is set to `-EPERM`.

Drivers must not return an error solely because the requested format doesn't match the device capabilities. They must instead modify the format to match what the hardware can provide. The modified format should be as close as possible to the original request.

```
    - ``which``
    - Format to modified, from enum
      :ref:`v4l2_subdev_format_whence <v4l2-subdev-format-whence>`.
  * - struct :c:type:`v4l2_mbus_framefmt`
    - ``format``
    - Definition of an image format, see :c:type:`v4l2_mbus_framefmt` for
      details.
  * - __u32
    - ``reserved``\ [8]
    - Reserved for future extensions. Applications and drivers must set
      the array to zero.
```

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.

EBUSY

> The format can't be changed because the pad is currently busy. This can be caused, for instance, by an active video stream on the pad. The ioctl must not be retried without performing another action to fix the problem first. Only returned by `VIDIOC_SUBDEV_S_FMT`

EINVAL

> The struct :c:type:`v4l2_subdev_format` `pad` references a non-existing pad, or the `which` field references a non-existing format.

EPERM

The `VIDIOC_SUBDEV_S_FMT` ioctl has been called on a read-only subdevice and the `which` field is set to `V4L2_SUBDEV_FORMAT_ACTIVE`.

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.