

# Streaming I/O (DMA buffer importing)

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 2)

Unknown directive type "c:namespace".

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

The DMABUF framework provides a generic method for sharing buffers between multiple devices. Device drivers that support DMABUF can export a DMA buffer to userspace as a file descriptor (known as the exporter role), import a DMA buffer from userspace using a file descriptor previously exported for a different or the same device (known as the importer role), or both. This section describes the DMABUF importer role API in V4L2.

Refer to [ref:DMABUF exporting <VIDIOC\\_EXPBUF>](#) for details about exporting V4L2 buffers as DMABUF file descriptors.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 18); [backlink](#)

Unknown interpreted text role "ref".

Input and output devices support the streaming I/O method when the `V4L2_CAP_STREAMING` flag in the `capabilities` field of struct `:c:type:`v4l2_capability`` returned by the `ref:VIDIOC_QUERYCAP <VIDIOC_QUERYCAP>` ioctl is set. Whether importing DMA buffers through DMABUF file descriptors is supported is determined by calling the `ref:VIDIOC_REQBUFS <VIDIOC_REQBUFS>` ioctl with the memory type set to `V4L2_MEMORY_DMABUF`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 21); [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] dmabuf.rst, line 21); [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] dmabuf.rst, line 21); [backlink](#)

Unknown interpreted text role "ref".

This I/O method is dedicated to sharing DMA buffers between different devices, which may be V4L devices or other video-related devices (e.g. DRM). Buffers (planes) are allocated by a driver on behalf of an application. Next, these buffers are exported to the application as file descriptors using an API which is specific for an allocator driver. Only such file descriptor are exchanged. The descriptors and meta-information are passed in struct `:c:type:`v4l2_buffer`` (or in struct `:c:type:`v4l2_plane`` in the multi-planar API case). The driver must be switched into DMABUF I/O mode by calling the `ref:VIDIOC_REQBUFS <VIDIOC_REQBUFS>` with the desired buffer 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] dmabuf.rst, line 29); [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] dmabuf.rst, line 29); [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] dmabuf.rst, line 29); [backlink](#)

Unknown interpreted text role "ref".

## Example: Initiating streaming I/O with DMABUF file descriptors

```
struct v4l2_requestbuffers reqbuf;

memset(&reqbuf, 0, sizeof (reqbuf));
reqbuf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
reqbuf.memory = V4L2_MEMORY_DMABUF;
reqbuf.count = 1;

if (ioctl(fd, VIDIOC_REQBUFS, &reqbuf) == -1) {
    if (errno == EINVAL)
        printf("Video capturing or DMABUF streaming is not supported\n");
    else
        perror("VIDIOC_REQBUFS");

    exit(EXIT_FAILURE);
}
```

The buffer (plane) file descriptor is passed on the fly with the `ref`VIDIOC_QBUF <VIDIOC_QBUF>`` ioctl. In case of multiplanar buffers, every plane can be associated with a different DMABUF descriptor. Although buffers are commonly cycled, applications can pass a different DMABUF descriptor at each `ref`VIDIOC_QBUF <VIDIOC_QBUF>`` call.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 61); [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] dmabuf.rst, line 61); [backlink](#)

Unknown interpreted text role "ref".

## Example: Queueing DMABUF using single plane API

```
int buffer_queue(int v4lfd, int index, int dmafd)
{
    struct v4l2_buffer buf;

    memset(&buf, 0, sizeof buf);
    buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
    buf.memory = V4L2_MEMORY_DMABUF;
    buf.index = index;
    buf.m.fd = dmafd;

    if (ioctl(v4lfd, VIDIOC_QBUF, &buf) == -1) {
        perror("VIDIOC_QBUF");
        return -1;
    }

    return 0;
}
```

## Example 3.6. Queueing DMABUF using multi plane API

```
int buffer_queue_mp(int v4lfd, int index, int dmafd[], int n_planes)
{
    struct v4l2_buffer buf;
    struct v4l2_plane planes[VIDEO_MAX_PLANES];
    int i;

    memset(&buf, 0, sizeof buf);
    buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE_MPLANE;
    buf.memory = V4L2_MEMORY_DMABUF;
```

```

buf.index = index;
buf.m.planes = planes;
buf.length = n_planes;

memset(&planes, 0, sizeof planes);

for (i = 0; i < n_planes; ++i)
    buf.m.planes[i].m.fd = dmafd[i];

if (ioctl(v4lfd, VIDIOC_QBUF, &buf) == -1) {
    perror("VIDIOC_QBUF");
    return -1;
}

return 0;
}

```

Captured or displayed buffers are dequeued with the `ref`VIDIOC_DQBUF <VIDIOC_QBUF>`` ioctl. The driver can unlock the buffer at any time between the completion of the DMA and this ioctl. The memory is also unlocked when `ref`VIDIOC_STREAMOFF <VIDIOC_STREAMON>`` is called, `ref`VIDIOC_REQBUFS <VIDIOC_REQBUFS>``, or when the device is closed.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 121); [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] dmabuf.rst, line 121); [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] dmabuf.rst, line 121); [backlink](#)

Unknown interpreted text role "ref".

For capturing applications it is customary to enqueue a number of empty buffers, to start capturing and enter the read loop. Here the application waits until a filled buffer can be dequeued, and re-enqueues the buffer when the data is no longer needed. Output applications fill and enqueue buffers, when enough buffers are stacked up output is started. In the write loop, when the application runs out of free buffers it must wait until an empty buffer can be dequeued and reused. Two methods exist to suspend execution of the application until one or more buffers can be dequeued. By default `ref`VIDIOC_DQBUF <VIDIOC_QBUF>`` blocks when no buffer is in the outgoing queue. When the `O_NONBLOCK` flag was given to the `c:func:`open()`` function, `ref`VIDIOC_DQBUF <VIDIOC_QBUF>`` returns immediately with an `EAGAIN` error code when no buffer is available. The `c:func:`select()`` and `c:func:`poll()`` functions are always available.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\linux-master [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 128); [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] dmabuf.rst, line 128); [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] dmabuf.rst, line 128); [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] dmabuf.rst, line 128); [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] dmabuf.rst, line 128); [backlink](#)

Unknown interpreted text role "c:func".

To start and stop capturing or displaying applications call the `ref`VIDIOC_STREAMON <VIDIOC_STREAMON>`` and `ref`VIDIOC_STREAMOFF <VIDIOC_STREAMON>`` ioctls.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\[linux-master] [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 144); [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] dmabuf.rst, line 144); [backlink](#)

Unknown interpreted text role "ref".

#### Note

`ref`VIDIOC_STREAMOFF <VIDIOC_STREAMON>`` removes all buffers from both queues and unlocks all buffers as a side effect. Since there is no notion of doing anything "how" on a multitasking system, if an application needs to synchronize with another event it should examine the struct `c:type`v4l2_buffer`` timestamp of captured or outputted buffers.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\[linux-master] [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 150); [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] dmabuf.rst, line 150); [backlink](#)

Unknown interpreted text role "c:type".

Drivers implementing DMABUF importing I/O must support the `ref`VIDIOC_REQBUFS <VIDIOC_REQBUFS>``, `ref`VIDIOC_QBUF <VIDIOC_QBUF>``, `ref`VIDIOC_DQBUF <VIDIOC_QBUF>``, `ref`VIDIOC_STREAMON <VIDIOC_STREAMON>`` and `ref`VIDIOC_STREAMOFF <VIDIOC_STREAMON>`` ioctls, and the `c:func`select()`` and `c:func`poll()`` functions.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v4l\[linux-master] [Documentation] [userspace-api] [media] [v4l] dmabuf.rst, line 157); [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] dmabuf.rst, line 157); [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] [v41] dmabuf.rst, line 157); [backlink](#)

Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v41\[linux-master] [Documentation] [userspace-api] [media] [v41] dmabuf.rst, line 157); [backlink](#)

Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v41\[linux-master] [Documentation] [userspace-api] [media] [v41] dmabuf.rst, line 157); [backlink](#)

Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v41\[linux-master] [Documentation] [userspace-api] [media] [v41] dmabuf.rst, line 157); [backlink](#)

Unknown interpreted text role "c:func".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\v41\[linux-master] [Documentation] [userspace-api] [media] [v41] dmabuf.rst, line 157); [backlink](#)

Unknown interpreted text role "c:func".