ioctl DMX_EXPBUF

Name

DMX_EXPBUF - Export a buffer as a DMABUF file descriptor.

> **Warning**
>
> this API is still experimental

Synopsis

```
int ioctl(int fd, DMX_EXPBUF, struct dmx_exportbuffer *argp)
```

Arguments

fd

> File descriptor returned by :c:func:`open()`.
>

argp

> Pointer to struct :c:type:`dmx_exportbuffer`.
>

Description

This ioctl is an extension to the memory mapping I/O method. It can be used to export a buffer as a DMABUF file at any time after buffers have been allocated with the :ref:`DMX_REQBUFS` ioctl.

To export a buffer, applications fill struct :c:type:`dmx_exportbuffer`. Applications must set the `index` field. Valid index numbers range from zero to the number of buffers allocated with :ref:`DMX_REQBUFS` (struct :c:type:`dmx_requestbuffers` `count`) minus one. Additional flags may be posted in the `flags` field. Refer to a manual for open() for details. Currently only O_CLOEXEC, O_RDONLY, O_WRONLY, and O_RDWR are supported. All other fields must be set to zero. In the case of multi-planar API, every plane is exported separately using multiple :ref:`DMX_EXPBUF` calls.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\dvb\[linux-master][Documentation][userspace-api][media][dvb]dmx-expbuf.rst, line 40);` *backlink*
>
> Unknown interpreted text role "c:type".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\dvb\[linux-master][Documentation][userspace-api][media][dvb]dmx-expbuf.rst, line 40);` *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\dvb\[linux-master][Documentation][userspace-api][media][dvb]dmx-expbuf.rst, line 40);` *backlink*
>
> Unknown interpreted text role "c:type".

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\dvb\[linux-master][Documentation][userspace-api][media][dvb]dmx-expbuf.rst, line 40);` *backlink*
>
> Unknown interpreted text role "ref".

After calling :ref:`DMX_EXPBUF` the `fd` field will be set by a driver, on success. This is a DMABUF file descriptor. The application may pass it to other DMABUF-aware devices. It is recommended to close a DMABUF file when it is no longer used to allow the associated memory to be reclaimed.

> **System Message: ERROR/3 (**`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\dvb\[linux-master][Documentation][userspace-api][media][dvb]dmx-expbuf.rst, line 51);` *backlink*
>
> Unknown interpreted text role "ref".

Examples

```c
int buffer_export(int v4lfd, enum dmx_buf_type bt, int index, int *dmafd)
{
    struct dmx_exportbuffer expbuf;

    memset(&expbuf, 0, sizeof(expbuf));
    expbuf.type = bt;
    expbuf.index = index;
    if (ioctl(v4lfd, DMX_EXPBUF, &expbuf) == -1) {
        perror("DMX_EXPBUF");
        return -1;
    }

    *dmafd = expbuf.fd;

    return 0;
}
```

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\dvb\[linux-master][Documentation][userspace-api][media][dvb]dmx-expbuf.rst, line 81);` *backlink*

Unknown interpreted text role "ref".

EINVAL

A queue is not in MMAP mode or DMABUF exporting is not supported or `flags` or `index` fields are invalid.