

**NAME**

`ioctl_ficlonerange`, `ioctl_ficlone` – share some the data of one file with another file

**SYNOPSIS**

```
#include <sys/ioctl.h>
```

```
#include <linux/fs.h>
```

```
int ioctl(int dest_fd, FICLONERANGE, struct file_clone_range *arg);
```

```
int ioctl(int dest_fd, FICLONE, int src_fd);
```

**DESCRIPTION**

If a filesystem supports files sharing physical storage between multiple files ("reflink"), this `ioctl(2)` operation can be used to make some of the data in the `src_fd` file appear in the `dest_fd` file by sharing the underlying storage, which is faster than making a separate physical copy of the data. Both files must reside within the same filesystem. If a file write should occur to a shared region, the filesystem must ensure that the changes remain private to the file being written. This behavior is commonly referred to as "copy on write".

This `ioctl` reflinks up to `src_length` bytes from file descriptor `src_fd` at offset `src_offset` into the file `dest_fd` at offset `dest_offset`, provided that both are files. If `src_length` is zero, the `ioctl` reflinks to the end of the source file. This information is conveyed in a structure of the following form:

```
struct file_clone_range {
    __s64 src_fd;
    __u64 src_offset;
    __u64 src_length;
    __u64 dest_offset;
};
```

Clones are atomic with regards to concurrent writes, so no locks need to be taken to obtain a consistent cloned copy.

The **FICLONE** `ioctl` clones entire files.

**RETURN VALUE**

On error, `-1` is returned, and `errno` is set to indicate the error.

**ERRORS**

Error codes can be one of, but are not limited to, the following:

**EBADF**

`src_fd` is not open for reading; `dest_fd` is not open for writing or is open for append-only writes; or the filesystem which `src_fd` resides on does not support reflink.

**EINVAL**

The filesystem does not support reflinking the ranges of the given files. This error can also appear if either file descriptor represents a device, FIFO, or socket. Disk filesystems generally require the offset and length arguments to be aligned to the fundamental block size. XFS and Btrfs do not support overlapping reflink ranges in the same file.

**EISDIR**

One of the files is a directory and the filesystem does not support shared regions in directories.

**EOPNOTSUPP**

This can appear if the filesystem does not support reflinking either file descriptor, or if either file descriptor refers to special inodes.

**EPERM**

`dest_fd` is immutable.

**ETXTBSY**

One of the files is a swap file. Swap files cannot share storage.

**EXDEV**

*dest\_fd* and *src\_fd* are not on the same mounted filesystem.

**VERSIONS**

These ioctl operations first appeared in Linux 4.5. They were previously known as **BTRFS\_IOC\_CLONE** and **BTRFS\_IOC\_CLONE\_RANGE**, and were private to Btrfs.

**CONFORMING TO**

This API is Linux-specific.

**NOTES**

Because a copy-on-write operation requires the allocation of new storage, the **fallocate(2)** operation may unshare shared blocks to guarantee that subsequent writes will not fail because of lack of disk space.

**SEE ALSO**

**ioctl(2)**

**COLOPHON**

This page is part of release 5.05 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.