



Rocky Enterprise Linux 9.2 Manual Pages on command 'ioctl.2'

C:\>man ioctl.2

IOCTL(2) Linux Programmer's Manual IOCTL(2)

NAME

ioctl - control device

SYNOPSIS

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

```
int ioctl(int fd, unsigned long request, ...);
```

DESCRIPTION

The `ioctl()` system call manipulates the underlying device parameters of special files. In particular, many operating characteristics of character special files (e.g., terminals) may be controlled with `ioctl()` requests. The argument `fd` must be an open file descriptor.

The second argument is a device-dependent request code. The third argument is an untyped pointer to memory. It's traditionally `char *argp` (from the days before `void *` was valid C), and will be so named for this discussion.

An `ioctl()` request has encoded in it whether the argument is an in parameter or out parameter, and the size of the argument `argp` in bytes. Macros and defines used in specifying an `ioctl()` request are located in the file `<sys/ioctl.h>`.

RETURN VALUE

Usually, on success zero is returned. A few `ioctl()` requests use the return value as an output parameter and return a nonnegative value on success. On error, `-1` is returned, and `errno` is set appropriately.

ERRORS

EBADF fd is not a valid file descriptor.

EFAULT argp references an inaccessible memory area.

EINVAL request or argp is not valid.

ENOTTY fd is not associated with a character special device.

ENOTTY The specified request does not apply to the kind of object that the file descriptor fd references.

CONFORMING TO

No single standard. Arguments, returns, and semantics of `ioctl()` vary according to the device driver in question (the call is used as a catch-all for operations that don't cleanly fit the UNIX stream I/O model). See `ioctl_list(2)` for a list of many of the known `ioctl()` calls. The `ioctl()` system call appeared in Version 7 AT&T UNIX.

NOTES

In order to use this call, one needs an open file descriptor. Often the `open(2)` call has unwanted side effects, that can be avoided under Linux by giving it the `O_NONBLOCK` flag.

SEE ALSO

`execve(2)`, `fcntl(2)`, `ioctl_console(2)`, `ioctl_fat(2)`, `ioctl_ficlone(2)`,
`ioctl_fideduperange(2)`, `ioctl_getfsmmap(2)`, `ioctl_iflags(2)`, `ioctl_list(2)`,
`ioctl_ns(2)`, `ioctl_tty(2)`, `ioctl_userfaultfd(2)`, `open(2)`, `sd(4)`, `tty(4)`

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/>.