

NAME

disk - common disk interfaces

SYNOPSIS

device cd

DESCRIPTION

Common block device IOCTLs

All the block devices in the system should support these disk ioctl(2) commands defined here. Much of this information is also available via the geom(2) attributes.

IOCTLS

The following ioctl(2) calls apply to disk drives, and are defined in the *<sys/disk.h>* header file.

DIOCGSECTORSIZE	(u_int) Get the sector or block size of the device in bytes. The sector size is the smallest unit of data which can be transferred from this device. This is usually a power of 2 but it might not be (e.g. CDROM audio). Operations to block devices such as lseek(2), read(2), and write may only be performed at file offsets that are integral multiple of this size.
DIOCGMEDIASIZE	(off_t) Get the size of the entire device in bytes. This should be a multiple of the sector size.
DIOCGFWSECTORS	(u_int) Return the firmware's notion of number of sectors per track. This value is mostly used for compatibility with various ill designed disk label formats. Use this value only when absolutely required. Its interpretation and use is largely obsolete.
DIOCGFWHEADS	(u_int) Return the firmware's notion of number of heads per cylinder. This value is mostly used for compatibility with various ill designed disk label formats. Use this value only when absolutely required. Its interpretation and use is largely obsolete.
DIOCGFLUSH	Flush write cache of the device.
DIOCGDELETE	(off_t[2]) Mark data on the device as unused. The first element is the offset to start deleting. The second element is the length to delete. Providers may use this information to free storage or instruct storage devices the contents can be discarded.

DIOCGIDENT

(char[DISK_IDENT_SIZE]) Get the ident for this provider. Ident is a unique and fixed identifier for this provider. Ident's properties are as follow:

- preserved between reboots,
- preserved across a provider being detached/attached,
- provider's name can change - ident can't,
- ident value should not be based on on-disk metadata; in other words, copying whole data from one disk to another should not yield the same ident for the other disk,
- there can be more than one provider with the same ident, but only if they point at exactly the same physical storage, this is the case for multipathing for example,
- GEOM classes that consume a single provider and provide single provider, like geli(8), and gbde(8), the identifier should be formed by attaching that provider's class name to the ident of the underlying provider,
- ident is an NUL-terminated ASCII string (is printable),
- ident is optional and applications can't relay on its presence.

DIOCGPROVIDERNAME (char[MAXPATHLEN]) Store the provider name for the device in a buffer. The buffer must be at least MAXPATHLEN bytes long.

DIOCGSTRIPESIZE (off_t) Get the size of the device's optimal access block in bytes. This should be a multiple of the sector size.

DIOCGSTRIPEOFFSET (off_t) Get the offset of the first device's optimal access block in bytes. This should be a multiple of the sector size.

DIOCGPHYSPTH (char[MAXPATHLEN]) Get a string defining the physical path for a given provider. This has similar rules to ident, but is intended to uniquely identify the physical location of the device, not the current occupant of that location. The buffer must be at least MAXPATHLEN bytes long.

DIOCGATTR

(struct diocgattr_arg)

```

struct diocgattr_arg {
    char name[64];
    int len;
    union {
        char str[DISK_IDENT_SIZE];
        off_t off;
        int i;
        uint16_t u16;
    } value;
};

```

Get a geom attribute from the provider. Format of the returned data is specific to the attribute.

DIOCZONECMD

(struct disk_zone_arg) Send disk zone commands.

DIOCSKERNELDUMP

(struct diocskerneldump_arg) Enable/Disable the device for kernel core dumps.

DIOCGKERNELDUMP

(struct diocskerneldump_arg) Get current kernel netdump configuration details for a given index.

```

/*
 * Sentinel values for kda_index.
 *
 * If kda_index is KDA_REMOVE_ALL, all dump configurations are cleared.
 *
 * If kda_index is KDA_REMOVE_DEV, all dump configurations for the specified
 * device are cleared.
 *
 * If kda_index is KDA_REMOVE, only the specified dump configuration for the
 * given device is removed from the list of fallback dump configurations.
 *
 * If kda_index is KDA_APPEND, the dump configuration is added after all
 * existing dump configurations.
 *
 * Otherwise, the new configuration is inserted into the fallback dump list at
 * index 'kda_index'.
 */

```

```
#define KDA_REMOVE          UINT8_MAX
#define KDA_REMOVE_ALL      (UINT8_MAX - 1)
#define KDA_REMOVE_DEV      (UINT8_MAX - 2)
#define KDA_APPEND          (UINT8_MAX - 3)
struct diocskerneldump_arg {
    uint8_t      kda_index;
    uint8_t      kda_compression;
    uint8_t      kda_encryption;
    uint8_t      kda_key[KERNELDUMP_KEY_MAX_SIZE];
    uint32_t      kda_encryptedkeysize;
    uint8_t      *kda_encryptedkey;
    char          kda_iface[IFNAMSIZ];
    union kd_ip   kda_server;
    union kd_ip   kda_client;
    union kd_ip   kda_gateway;
    uint8_t      kda_af;
};
```

HISTORY

The manual page was written by M Warner Losh <imp@FreeBSD.org> from text largely derived from <*sys/disk.h*>.