

**NAME**

BIO\_s\_fd, BIO\_set\_fd, BIO\_get\_fd, BIO\_new\_fd - file descriptor BIO

**SYNOPSIS**

```
#include <openssl/bio.h>

BIO_METHOD * BIO_s_fd(void);

#define BIO_set_fd(b, fd, c) BIO_int_ctrl(b, BIO_C_SET_FD, c, fd)
#define BIO_get_fd(b, c) BIO_ctrl(b, BIO_C_GET_FD, 0, (char *)c)

BIO *BIO_new_fd(int fd, int close_flag);
```

**DESCRIPTION**

*BIO\_s\_fd()* returns the file descriptor BIO method. This is a wrapper round the platforms file descriptor routines such as *read()* and *write()*.

*BIO\_read()* and *BIO\_write()* read or write the underlying descriptor. *BIO\_puts()* is supported but *BIO\_gets()* is not.

If the close flag is set then then *close()* is called on the underlying file descriptor when the BIO is freed.

*BIO\_reset()* attempts to change the file pointer to the start of file using *lseek(fd, 0, 0)*.

*BIO\_seek()* sets the file pointer to position **ofs** from start of file using *lseek(fd, ofs, 0)*.

*BIO\_tell()* returns the current file position by calling *lseek(fd, 0, 1)*.

*BIO\_set\_fd()* sets the file descriptor of BIO **b** to **fd** and the close flag to **c**.

*BIO\_get\_fd()* places the file descriptor in **c** if it is not NULL, it also returns the file descriptor. If **c** is not NULL it should be of type (int \*).

*BIO\_new\_fd()* returns a file descriptor BIO using **fd** and **close\_flag**.

**NOTES**

The behaviour of *BIO\_read()* and *BIO\_write()* depends on the behavior of the platforms *read()* and *write()* calls on the descriptor. If the underlying file descriptor is in a non blocking mode then the BIO will behave in the manner described in the [BIO\\_read\(3\)](#) and [BIO\\_should\\_retry\(3\)](#) manual pages.

File descriptor BIOs should not be used for socket I/O. Use socket BIOs instead.

**RETURN VALUES**

*BIO\_s\_fd()* returns the file descriptor BIO method.

*BIO\_reset()* returns zero for success and -1 if an error occurred. *BIO\_seek()* and *BIO\_tell()* return the current file position or -1 if an error occurred. These values reflect the underlying *lseek()* behaviour.

*BIO\_set\_fd()* always returns 1.

*BIO\_get\_fd()* returns the file descriptor or -1 if the BIO has not been initialized.

*BIO\_new\_fd()* returns the newly allocated BIO or NULL if an error occurred.

**EXAMPLE**

This is a file descriptor BIO version of “Hello World”:

```
BIO *out;
out = BIO_new_fd(fileno(stdout), BIO_NOCLOSE);
BIO_printf(out, "Hello World\n");
BIO_free(out);
```

**SEE ALSO**

[BIO\\_seek\(3\)](#), [BIO\\_tell\(3\)](#), [BIO\\_reset\(3\)](#), [BIO\\_read\(3\)](#), [BIO\\_write\(3\)](#), [BIO\\_puts\(3\)](#), [BIO\\_gets\(3\)](#), [BIO\\_printf\(3\)](#), [BIO\\_set\\_close\(3\)](#), [BIO\\_get\\_close\(3\)](#)