

**NAME**

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

**SYNOPSIS**

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

```
const BIO_METHOD *BIO_s_fd(void);
```

```
int BIO_set_fd(BIO *b, int fd, int c);
```

```
int BIO_get_fd(BIO *b, int *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\_ex()** and **BIO\_write\_ex()** read or write the underlying descriptor. **BIO\_puts()** is supported but **BIO\_gets()** is not.

If the close flag is set 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 such as by using **lseek(fd, 0, 0)**.

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

**BIO\_tell()** returns the current file position such as 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 of BIO **b** in **c** if it is not NULL. It also returns the file descriptor.

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

**NOTES**

The behaviour of **BIO\_read\_ex()** and **BIO\_write\_ex()** 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\_ex(3)** and **BIO\_should\_retry(3)** manual pages.

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

**BIO\_set\_fd()** and **BIO\_get\_fd()** are implemented as macros.

## RETURN VALUES

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

**BIO\_set\_fd()** returns 1 on success or  $\leq 0$  for failure.

**BIO\_get\_fd()** returns the file descriptor or -1 if the BIO has not been initialized. It also returns zero and negative values if other error occurs.

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

## EXAMPLES

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\_ex(3)**, **BIO\_write\_ex(3)**, **BIO\_puts(3)**, **BIO\_gets(3)**, **BIO\_printf(3)**, **BIO\_set\_close(3)**, **BIO\_get\_close(3)**

## COPYRIGHT

Copyright 2000-2021 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.