

**NAME**

BIO\_s\_file, BIO\_new\_file, BIO\_new\_fp, BIO\_set\_fp, BIO\_get\_fp, BIO\_read\_filename, BIO\_write\_filename, BIO\_append\_filename, BIO\_rw\_filename - FILE bio

**SYNOPSIS**

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

BIO_METHOD * BIO_s_file(void);
BIO *BIO_new_file(const char *filename, const char *mode);
BIO *BIO_new_fp(FILE *stream, int flags);

BIO_set_fp(BIO *b, FILE *fp, int flags);
BIO_get_fp(BIO *b, FILE **fpp);

int BIO_read_filename(BIO *b, char *name)
int BIO_write_filename(BIO *b, char *name)
int BIO_append_filename(BIO *b, char *name)
int BIO_rw_filename(BIO *b, char *name)
```

**DESCRIPTION**

*BIO\_s\_file()* returns the BIO file method. As its name implies it is a wrapper round the stdio FILE structure and it is a source/sink BIO.

Calls to *BIO\_read()* and *BIO\_write()* read and write data to the underlying stream. *BIO\_gets()* and *BIO\_puts()* are supported on file BIOs.

*BIO\_flush()* on a file BIO calls the *fflush()* function on the wrapped stream.

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

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

*BIO\_eof()* calls *feof()*.

Setting the BIO\_CLOSE flag calls *fclose()* on the stream when the BIO is freed.

*BIO\_new\_file()* creates a new file BIO with mode **mode** the meaning of **mode** is the same as the stdio function *fopen()*. The BIO\_CLOSE flag is set on the returned BIO.

*BIO\_new\_fp()* creates a file BIO wrapping **stream**. Flags can be: BIO\_CLOSE, BIO\_NOCLOSE (the close flag) BIO\_FP\_TEXT (sets the underlying stream to text mode, default is binary: this only has any effect under Win32).

*BIO\_set\_fp()* set the fp of a file BIO to **fp**. **flags** has the same meaning as in *BIO\_new\_fp()*, it is a macro.

*BIO\_get\_fp()* retrieves the fp of a file BIO, it is a macro.

*BIO\_seek()* is a macro that sets the position pointer to **offset** bytes from the start of file.

*BIO\_tell()* returns the value of the position pointer.

*BIO\_read\_filename()*, *BIO\_write\_filename()*, *BIO\_append\_filename()* and *BIO\_rw\_filename()* set the file BIO **b** to use file **name** for reading, writing, append or read write respectively.

**NOTES**

When wrapping stdout, stdin or stderr the underlying stream should not normally be closed so the BIO\_NOCLOSE flag should be set.

Because the file BIO calls the underlying stdio functions any quirks in stdio behaviour will be mirrored by the corresponding BIO.

On Windows *BIO\_new\_files* reserves for the filename argument to be UTF-8 encoded. In other words if you have to make it work in multi-lingual environment, encode file names in UTF-8.

**EXAMPLES**

File BIO “hello world”:

```
BIO *bio_out;
bio_out = BIO_new_fp(stdout, BIO_NOCLOSE);
BIO_printf(bio_out, "Hello World\n");
```

Alternative technique:

```
BIO *bio_out;
bio_out = BIO_new(BIO_s_file());
if(bio_out == NULL) /* Error ... */
if(!BIO_set_fp(bio_out, stdout, BIO_NOCLOSE)) /* Error ... */
BIO_printf(bio_out, "Hello World\n");
```

Write to a file:

```
BIO *out;
out = BIO_new_file("filename.txt", "w");
if(!out) /* Error occurred */
BIO_printf(out, "Hello World\n");
BIO_free(out);
```

Alternative technique:

```
BIO *out;
out = BIO_new(BIO_s_file());
if(out == NULL) /* Error ... */
if(!BIO_write_filename(out, "filename.txt")) /* Error ... */
BIO_printf(out, "Hello World\n");
BIO_free(out);
```

**RETURN VALUES**

*BIO\_s\_file()* returns the file BIO method.

*BIO\_new\_file()* and *BIO\_new\_fp()* return a file BIO or NULL if an error occurred.

*BIO\_set\_fp()* and *BIO\_get\_fp()* return 1 for success or 0 for failure (although the current implementation never return 0).

*BIO\_seek()* returns the same value as the underlying *fseek()* function: 0 for success or -1 for failure.

*BIO\_tell()* returns the current file position.

*BIO\_read\_filename()*, *BIO\_write\_filename()*, *BIO\_append\_filename()* and *BIO\_rw\_filename()* return 1 for success or 0 for failure.

**BUGS**

*BIO\_reset()* and *BIO\_seek()* are implemented using *fseek()* on the underlying stream. The return value for *fseek()* is 0 for success or -1 if an error occurred this differs from other types of BIO which will typically return 1 for success and a non positive value if an error occurred.

**SEE ALSO**

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