

**NAME**

BIO\_ctrl, BIO\_callback\_ctrl, BIO\_ptr\_ctrl, BIO\_int\_ctrl, BIO\_reset, BIO\_seek, BIO\_tell, BIO\_flush, BIO\_eof, BIO\_set\_close, BIO\_get\_close, BIO\_pending, BIO\_wpending, BIO\_ctrl\_pending, BIO\_ctrl\_wpending, BIO\_get\_info\_callback, BIO\_set\_info\_callback - BIO control operations

**SYNOPSIS**

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

```
long BIO_ctrl(BIO *bp, int cmd, long larg, void *parg);
long BIO_callback_ctrl(BIO *b, int cmd, void (*fp)(struct bio_st *, int, const c
char * BIO_ptr_ctrl(BIO *bp, int cmd, long larg);
long BIO_int_ctrl(BIO *bp, int cmd, long larg, int iarg);
```

```
int BIO_reset(BIO *b);
int BIO_seek(BIO *b, int ofs);
int BIO_tell(BIO *b);
int BIO_flush(BIO *b);
int BIO_eof(BIO *b);
int BIO_set_close(BIO *b, long flag);
int BIO_get_close(BIO *b);
int BIO_pending(BIO *b);
int BIO_wpending(BIO *b);
size_t BIO_ctrl_pending(BIO *b);
size_t BIO_ctrl_wpending(BIO *b);
```

```
int BIO_get_info_callback(BIO *b, bio_info_cb **cbp);
int BIO_set_info_callback(BIO *b, bio_info_cb *cb);
```

```
typedef void bio_info_cb(BIO *b, int oper, const char *ptr, int arg1, long arg2,
```

**DESCRIPTION**

*BIO\_ctrl()*, *BIO\_callback\_ctrl()*, *BIO\_ptr\_ctrl()* and *BIO\_int\_ctrl()* are BIO “control” operations taking arguments of various types. These functions are not normally called directly, various macros are used instead. The standard macros are described below, macros specific to a particular type of BIO are described in the specific BIOs manual page as well as any special features of the standard calls.

*BIO\_reset()* typically resets a BIO to some initial state, in the case of file related BIOs for example it rewinds the file pointer to the start of the file.

*BIO\_seek()* resets a file related BIO’s (that is file descriptor and FILE BIOs) file position pointer to **ofs** bytes from start of file.

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

*BIO\_flush()* normally writes out any internally buffered data, in some cases it is used to signal EOF and that no more data will be written.

*BIO\_eof()* returns 1 if the BIO has read EOF, the precise meaning of “EOF” varies according to the BIO type.

*BIO\_set\_close()* sets the BIO **b** close flag to **flag**. **flag** can take the value BIO\_CLOSE or BIO\_NOCLOSE. Typically BIO\_CLOSE is used in a source/sink BIO to indicate that the underlying I/O stream should be closed when the BIO is freed.

*BIO\_get\_close()* returns the BIOs close flag.

*BIO\_pending()*, *BIO\_ctrl\_pending()*, *BIO\_wpending()* and *BIO\_ctrl\_wpending()* return the number of pending characters in the BIOs read and write buffers. Not all BIOs support these calls. *BIO\_ctrl\_pending()* and *BIO\_ctrl\_wpending()* return a size\_t type and are functions, *BIO\_pending()* and *BIO\_wpending()* are macros which call *BIO\_ctrl()*.

## RETURN VALUES

*BIO\_reset()* normally returns 1 for success and 0 or -1 for failure. File BIOs are an exception, they return 0 for success and -1 for failure.

*BIO\_seek()* and *BIO\_tell()* both return the current file position on success and -1 for failure, except file BIOs which for *BIO\_seek()* always return 0 for success and -1 for failure.

*BIO\_flush()* returns 1 for success and 0 or -1 for failure.

*BIO\_eof()* returns 1 if EOF has been reached 0 otherwise.

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

*BIO\_get\_close()* returns the close flag value: BIO\_CLOSE or BIO\_NOCLOSE.

*BIO\_pending()*, *BIO\_ctrl\_pending()*, *BIO\_wpending()* and *BIO\_ctrl\_wpending()* return the amount of pending data.

## NOTES

*BIO\_flush()*, because it can write data may return 0 or -1 indicating that the call should be retried later in a similar manner to *BIO\_write()*. The *BIO\_should\_retry()* call should be used and appropriate action taken is the call fails.

The return values of *BIO\_pending()* and *BIO\_wpending()* may not reliably determine the amount of pending data in all cases. For example in the case of a file BIO some data may be available in the FILE structures internal buffers but it is not possible to determine this in a portably way. For other types of BIO they may not be supported.

Filter BIOs if they do not internally handle a particular *BIO\_ctrl()* operation usually pass the operation to the next BIO in the chain. This often means there is no need to locate the required BIO for a particular operation, it can be called on a chain and it will be automatically passed to the relevant BIO. However this can cause unexpected results: for example no current filter BIOs implement *BIO\_seek()*, but this may still succeed if the chain ends in a FILE or file descriptor BIO.

Source/sink BIOs return an 0 if they do not recognize the *BIO\_ctrl()* operation.

## BUGS

Some of the return values are ambiguous and care should be taken. In particular a return value of 0 can be returned if an operation is not supported, if an error occurred, if EOF has not been reached and in the case of *BIO\_seek()* on a file BIO for a successful operation.

## SEE ALSO

TBA