

**NAME**

BIO\_f\_buffer - buffering BIO

**SYNOPSIS**

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

```
BIO_METHOD * BIO_f_buffer(void);
```

```
#define BIO_get_buffer_num_lines(b) BIO_ctrl(b,BIO_C_GET_BUFF_NUM_LINES,0,NULL)
#define BIO_set_read_buffer_size(b,size) BIO_int_ctrl(b,BIO_C_SET_BUFF_SIZE,size,0)
#define BIO_set_write_buffer_size(b,size) BIO_int_ctrl(b,BIO_C_SET_BUFF_SIZE,size,1)
#define BIO_set_buffer_size(b,size) BIO_ctrl(b,BIO_C_SET_BUFF_SIZE,size,NULL)
#define BIO_set_buffer_read_data(b,buf,num) BIO_ctrl(b,BIO_C_SET_BUFF_READ_DATA,num,buf)
```

**DESCRIPTION**

*BIO\_f\_buffer()* returns the buffering BIO method.

Data written to a buffering BIO is buffered and periodically written to the next BIO in the chain. Data read from a buffering BIO comes from an internal buffer which is filled from the next BIO in the chain. Both *BIO\_gets()* and *BIO\_puts()* are supported.

Calling *BIO\_reset()* on a buffering BIO clears any buffered data.

*BIO\_get\_buffer\_num\_lines()* returns the number of lines currently buffered.

*BIO\_set\_read\_buffer\_size()*, *BIO\_set\_write\_buffer\_size()* and *BIO\_set\_buffer\_size()* set the read, write or both read and write buffer sizes to **size**. The initial buffer size is `DEFAULT_BUFFER_SIZE`, currently 4096. Any attempt to reduce the buffer size below `DEFAULT_BUFFER_SIZE` is ignored. Any buffered data is cleared when the buffer is resized.

*BIO\_set\_buffer\_read\_data()* clears the read buffer and fills it with **num** bytes of **buf**. If **num** is larger than the current buffer size the buffer is expanded.

**NOTES**

Buffering BIOs implement *BIO\_gets()* by using *BIO\_read()* operations on the next BIO in the chain. By prepending a buffering BIO to a chain it is therefore possible to provide *BIO\_gets()* functionality if the following BIOs do not support it (for example SSL BIOs).

Data is only written to the next BIO in the chain when the write buffer fills or when *BIO\_flush()* is called. It is therefore important to call *BIO\_flush()* whenever any pending data should be written such as when removing a buffering BIO using *BIO\_pop()*. *BIO\_flush()* may need to be retried if the ultimate source/sink BIO is non blocking.

**RETURN VALUES**

*BIO\_f\_buffer()* returns the buffering BIO method.

*BIO\_get\_buffer\_num\_lines()* returns the number of lines buffered (may be 0).

*BIO\_set\_read\_buffer\_size()*, *BIO\_set\_write\_buffer\_size()* and *BIO\_set\_buffer\_size()* return 1 if the buffer was successfully resized or 0 for failure.

*BIO\_set\_buffer\_read\_data()* returns 1 if the data was set correctly or 0 if there was an error.

**SEE ALSO**

[BIO\(3\)](#), [BIO\\_reset\(3\)](#), [BIO\\_flush\(3\)](#), [BIO\\_pop\(3\)](#), [BIO\\_ctrl\(3\)](#), [BIO\\_int\\_ctrl\(3\)](#)