



*Full credit is given to the above companies including the OS that this PDF file was generated!*

***Rocky Enterprise Linux 9.2 Manual Pages on command 'BIO\_set\_read\_buffer\_size.3ossl'***

***\$ man BIO\_set\_read\_buffer\_size.3ossl***

BIO\_F\_BUFFER(3ossl)            OpenSSL            BIO\_F\_BUFFER(3ossl)

**NAME**

BIO\_get\_buffer\_num\_lines, BIO\_set\_read\_buffer\_size,  
BIO\_set\_write\_buffer\_size, BIO\_set\_buffer\_size,  
BIO\_set\_buffer\_read\_data, BIO\_f\_buffer - buffering BIO

**SYNOPSIS**

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

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

```
long BIO_get_buffer_num_lines(BIO *b);
```

```
long BIO_set_read_buffer_size(BIO *b, long size);
```

```
long BIO_set_write_buffer_size(BIO *b, long size);
```

```
long BIO_set_buffer_size(BIO *b, long size);
```

```
long BIO_set_buffer_read_data(BIO *b, void *buf, long num);
```

## 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

These functions, other than `BIO_f_buffer()`, are implemented as macros.

Buffering BIOs implement `BIO_read_ex()` and `BIO_gets()` by using `BIO_read_ex()` operations on the next BIO in the chain and storing the result in an internal buffer, from which bytes are given back to the caller as appropriate for the call; a `BIO_gets()` is guaranteed to give the caller a whole line, and `BIO_read_ex()` is guaranteed to give the caller the number of bytes it asks for, unless there's an error or end

of communication is reached in the next BIO. By prepending a buffering BIO to a chain it is therefore possible to provide BIO\_gets() or exact size BIO\_read\_ex() functionality if the following BIOs do not support it.

Do not add more than one BIO\_f\_buffer() to a BIO chain. The result of doing so will force a full read of the size of the internal buffer of the top BIO\_f\_buffer(), which is 4 KiB at a minimum.

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) or a negative value in case of errors.

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(7), BIO\_reset(3), BIO\_flush(3), BIO\_pop(3), BIO\_ctrl(3).

## COPYRIGHT

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>.

3.0.7                    2023-07-13                    BIO\_F\_BUFFER(3ossl)