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

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'BIO_f_buffer.3ossl' command

$ man BIO_f_buffer.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