



*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\_set\_next.3ossl' command***

***\$ man BIO\_set\_next.3ossl***

BIO\_PUSH(3ossl)                      OpenSSL                      BIO\_PUSH(3ossl)

### NAME

BIO\_push, BIO\_pop, BIO\_set\_next - add and remove BIOs from a chain

### SYNOPSIS

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

```
BIO *BIO_push(BIO *b, BIO *next);
```

```
BIO *BIO_pop(BIO *b);
```

```
void BIO_set_next(BIO *b, BIO *next);
```

### DESCRIPTION

BIO\_push() pushes b on next. If b is NULL the function does nothing and returns next. Otherwise it prepends b, which may be a single BIO or a chain of BIOs, to next (unless next is NULL). It then makes a control call on b and returns b.

BIO\_pop() removes the BIO b from any chain it is part of. If b is NULL the function does nothing and returns NULL. Otherwise it makes a control call on b and returns the next BIO in the chain, or NULL if there is no next BIO. The removed BIO becomes a single BIO with no association with the original chain, it can thus be freed or be made

part of a different chain.

`BIO_set_next()` replaces the existing next BIO in a chain with the BIO pointed to by `next`. The new chain may include some of the same BIOs from the old chain or it may be completely different.

## NOTES

The names of these functions are perhaps a little misleading.

`BIO_push()` joins two BIO chains whereas `BIO_pop()` deletes a single BIO from a chain, the deleted BIO does not need to be at the end of a chain.

The process of calling `BIO_push()` and `BIO_pop()` on a BIO may have additional consequences (a control call is made to the affected BIOs).

Any effects will be noted in the descriptions of individual BIOs.

## RETURN VALUES

`BIO_push()` returns the head of the chain, which usually is `b`, or `next` if `b` is `NULL`.

`BIO_pop()` returns the next BIO in the chain, or `NULL` if there is no next BIO.

## EXAMPLES

For these examples suppose `md1` and `md2` are digest BIOs, `b64` is a base64 BIO and `f` is a file BIO.

If the call:

```
BIO_push(b64, f);
```

is made then the new chain will be `b64-f`. After making the calls

```
BIO_push(md2, b64);
```

```
BIO_push(md1, md2);
```

the new chain is md1-md2-b64-f. Data written to md1 will be digested by md1 and md2, base64 encoded, and finally written to f.

It should be noted that reading causes data to pass in the reverse direction, that is data is read from f, base64 decoded, and digested by md2 and then md1.

The call:

```
BIO_pop(md2);
```

will return b64 and the new chain will be md1-b64-f. Data can be written to and read from md1 as before, except that md2 will no more be applied.

## SEE ALSO

bio(7)

## HISTORY

The `BIO_set_next()` function was added in OpenSSL 1.1.0.

## COPYRIGHT

Copyright 2000-2021 The OpenSSL Project Authors. All Rights Reserved.

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>.