## NAME

BIO_push, BIO_pop - add and remove BIOs from a chain.

## SYNOPSIS

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

BIO * BIO_push(BIO *b,BIO *append);
BIO * BIO_pop(BIO *b);
```

## DESCRIPTION

The *BIO_push()* function appends the BIO **append** to **b**, it returns **b**.

*BIO_pop()* removes the BIO **b** from a chain and returns the next BIO in the chain, or NULL if there is no next BIO. The removed BIO then becomes a single BIO with no association with the original chain, it can thus be freed or attached to a different chain.

## 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.

## 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 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 **md1** and **md2**. If the call:

```
BIO_pop(md2);
```

The call will return **b64** and the new chain will be **md1-b64-f** data can be written to **md1** as before.

## RETURN VALUES

*BIO_push()* returns the end of the chain, **b**.

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

## SEE ALSO

TBA