

## NAME

BIO\_new, BIO\_set, BIO\_free, BIO\_vfree, BIO\_free\_all - BIO allocation and freeing functions

## SYNOPSIS

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

BIO * BIO_new(BIO_METHOD *type);
int BIO_set(BIO *a,BIO_METHOD *type);
int BIO_free(BIO *a);
void BIO_vfree(BIO *a);
void BIO_free_all(BIO *a);
```

## DESCRIPTION

The *BIO\_new()* function returns a new BIO using method **type**.

*BIO\_set()* sets the method of an already existing BIO.

*BIO\_free()* frees up a single BIO, *BIO\_vfree()* also frees up a single BIO but it does not return a value. Calling *BIO\_free()* may also have some effect on the underlying I/O structure, for example it may close the file being referred to under certain circumstances. For more details see the individual BIO\_METHOD descriptions.

*BIO\_free\_all()* frees up an entire BIO chain, it does not halt if an error occurs freeing up an individual BIO in the chain.

## RETURN VALUES

*BIO\_new()* returns a newly created BIO or NULL if the call fails.

*BIO\_set()*, *BIO\_free()* return 1 for success and 0 for failure.

*BIO\_free\_all()* and *BIO\_vfree()* do not return values.

## NOTES

Some BIOs (such as memory BIOs) can be used immediately after calling *BIO\_new()*. Others (such as file BIOs) need some additional initialization, and frequently a utility function exists to create and initialize such BIOs.

If *BIO\_free()* is called on a BIO chain it will only free one BIO resulting in a memory leak.

Calling *BIO\_free\_all()* a single BIO has the same effect as calling *BIO\_free()* on it other than the discarded return value.

Normally the **type** argument is supplied by a function which returns a pointer to a BIO\_METHOD. There is a naming convention for such functions: a source/sink BIO is normally called *BIO\_s\_\**() and a filter BIO *BIO\_f\_\**();

## EXAMPLE

Create a memory BIO:

```
BIO *mem = BIO_new(BIO_s_mem());
```

## SEE ALSO

TBA