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