NAME

BIO_set_callback,   BIO_get_callback,   BIO_set_callback_arg,   BIO_get_callback_arg,
BIO_debug_callback - BIO callback functions

SYNOPSIS

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

#define BIO_set_callback(b,cb) ((b)->callback=(cb))
#define BIO_get_callback(b) ((b)->callback)
#define BIO_set_callback_arg(b,arg) ((b)->cb_arg=(char *)(arg))
#define BIO_get_callback_arg(b) ((b)->cb_arg)

long BIO_debug_callback(BIO *bio,int cmd,const char *argp,int argi,
long argl,long ret);

typedef long (*callback)(BIO *b, int oper, const char *argp,
int argi, long argl, long retvalue);
```

DESCRIPTION

BIO_set_callback() and *BIO_get_callback()* set and retrieve the BIO callback, they are both
macros. The callback is called during most high level BIO operations. It can be used for debugging
purposes to trace operations on a BIO or to modify its operation.

BIO_set_callback_arg() and *BIO_get_callback_arg()* are macros which can be used to set and
retrieve an argument for use in the callback.

BIO_debug_callback() is a standard debugging callback which prints out information relating to
each BIO operation. If the callback argument is set if is interpreted as a BIO to send the
information to, otherwise stderr is used.

callback() is the callback function itself. The meaning of each argument is described below.

The BIO the callback is attached to is passed in **b**.

oper is set to the operation being performed. For some operations the callback is called twice,
once before and once after the actual operation, the latter case has **oper** or'ed with
BIO_CB_RETURN.

The meaning of the arguments **argp**, **argi** and **argl** depends on the value of **oper**, that is the
operation being performed.

retvalue is the return value that would be returned to the application if no callback were
present. The actual value returned is the return value of the callback itself. In the case of
callbacks called before the actual BIO operation 1 is placed in retvalue, if the return value is not
positive it will be immediately returned to the application and the BIO operation will not be
performed.

The callback should normally simply return **retvalue** when it has finished processing, unless if
specifically wishes to modify the value returned to the application.

CALLBACK OPERATIONS

BIO_free(b)
callback(b, BIO_CB_FREE, NULL, 0L, 0L, 1L) is called before the free operation.

BIO_read(b, out, outl)
callback(b, BIO_CB_READ, out, outl, 0L, 1L) is called before the read and callback(b,
BIO_CB_READ|BIO_CB_RETURN, out, outl, 0L, retvalue) after.

BIO_write(b, in, inl)
callback(b, BIO_CB_WRITE, in, inl, 0L, 1L) is called before the write and callback(b,
BIO_CB_WRITE|BIO_CB_RETURN, in, inl, 0L, retvalue) after.

BIO_gets(b, out, outl)
> callback(b, BIO_CB_GETS, out, outl, 0L, 1L) is called before the operation and callback(b, BIO_CB_GETS|BIO_CB_RETURN, out, outl, 0L, retvalue) after.

BIO_puts(b, in)
> callback(b, BIO_CB_WRITE, in, 0, 0L, 1L) is called before the operation and callback(b, BIO_CB_WRITE|BIO_CB_RETURN, in, 0, 0L, retvalue) after.

BIO_ctrl(BIO *b, int cmd, long larg, void *parg)
> callback(b,BIO_CB_CTRL,parg,cmd,larg,1L) is called before the call and callback(b,BIO_CB_CTRL|BIO_CB_RETURN,parg,cmd, larg,ret) after.

EXAMPLE
The *BIO_debug_callback()* function is a good example, its source is in crypto/bio/bio_cb.c

SEE ALSO
TBA