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_get_fd.3ossl' command

$ man BIO_get_fd.3ossl

BIO_S_FD(3ossl) OpenSSL BIO_S_FD(3ossl)

NAME

BIO_s_fd, BIO_set_fd, BIO_get_fd, BIO_new_fd - file descriptor BIO

SYNOPSIS

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

const BIO_METHOD *BIO_s_fd(void);

int BIO_set_fd(BIO *b, int fd, int c);

int BIO_get_fd(BIO *b, int *c);

BIO *BIO_new_fd(int fd, int close_flag);
```

DESCRIPTION

BIO_s_fd() returns the file descriptor BIO method. This is a wrapper round the platforms file descriptor routines such as read() and write().

BIO_read_ex() and BIO_write_ex() read or write the underlying descriptor. BIO_puts() is supported but BIO_gets() is not.

If the close flag is set then close() is called on the underlying file descriptor when the BIO is freed.

BIO_reset() attempts to change the file pointer to the start of file such as by using lseek(fd, 0, 0).

BIO_seek() sets the file pointer to position ofs from start of file such as by using lseek(fd, ofs, 0).

BIO_tell() returns the current file position such as by calling lseek(fd, 0, 1).

BIO_set_fd() sets the file descriptor of BIO b to fd and the close flag to c.

BIO_get_fd() places the file descriptor of BIO b in c if it is not NULL. It also returns the file descriptor.

BIO_new_fd() returns a file descriptor BIO using fd and close_flag.

NOTES

The behaviour of BIO_read_ex() and BIO_write_ex() depends on the behavior of the platforms read() and write() calls on the descriptor. If the underlying file descriptor is in a non blocking mode then the BIO will behave in the manner described in the BIO_read_ex(3) and BIO_should_retry(3) manual pages.

File descriptor BIOs should not be used for socket I/O. Use socket BIOs instead.

BIO_set_fd() and BIO_get_fd() are implemented as macros.

RETURN VALUES

BIO_s_fd() returns the file descriptor BIO method.

BIO_set_fd() returns 1 on success or <=0 for failure.

BIO_get_fd() returns the file descriptor or -1 if the BIO has not been initialized. It also returns zero and negative values if other error occurs.

BIO_new_fd() returns the newly allocated BIO or NULL is an error occurred.

EXAMPLES

This is a file descriptor BIO version of "Hello World":

```
BIO *out;
out = BIO_new_fd(fileno(stdout), BIO_NOCLOSE);
BIO_printf(out, "Hello World\n");
BIO_free(out);
```

SEE ALSO

BIO_seek(3), BIO_tell(3), BIO_reset(3), BIO_read_ex(3), BIO_write_ex(3), BIO_puts(3), BIO_gets(3), BIO_printf(3), BIO_set_close(3), BIO_get_close(3)

COPYRIGHT