NAME

BIO_s_file, BIO_new_file, BIO_new_fp, BIO_set_fp, BIO_get_fp, BIO_read_filename, BIO_write_filename, BIO_append_filename, BIO_rw_filename - FILE bio

SYNOPSIS

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

const BIO_METHOD *BIO_s_file(void);
BIO *BIO_new_file(const char *filename, const char *mode);
BIO *BIO_new_fp(FILE *stream, int flags);

BIO_set_fp(BIO *b, FILE *fp, int flags);
BIO_get_fp(BIO *b, FILE **fpp);

int BIO_read_filename(BIO *b, char *name);
int BIO_write_filename(BIO *b, char *name);
int BIO_append_filename(BIO *b, char *name);
int BIO_rw_filename(BIO *b, char *name);
```

DESCRIPTION

BIO_s_file() returns the BIO file method. As its name implies it is a wrapper round the stdio FILE structure and it is a source/sink BIO.

Calls to **BIO_read_ex()** and **BIO_write_ex()** read and write data to the underlying stream. **BIO_gets()** and **BIO_puts()** are supported on file BIOs.

BIO_flush() on a file BIO calls the **fflush()** function on the wrapped stream.

BIO_reset() attempts to change the file pointer to the start of file using fseek(stream, 0, 0).

BIO_seek() sets the file pointer to position **ofs** from start of file using fseek(stream, ofs, 0).

BIO_eof() calls **feof()**.

Setting the BIO_CLOSE flag calls **fclose()** on the stream when the BIO is freed.

BIO_new_file() creates a new file BIO with mode **mode** the meaning of **mode** is the same as the stdio function **fopen()**. The BIO_CLOSE flag is set on the returned BIO.

BIO_new_fp() creates a file BIO wrapping **stream**. Flags can be: BIO_CLOSE, BIO_NOCLOSE (the

close flag) BIO_FP_TEXT (sets the underlying stream to text mode, default is binary: this only has any effect under Win32).

BIO_set_fp() sets the fp of a file BIO to **fp**. **flags** has the same meaning as in **BIO_new_fp()**, it is a macro.

BIO_get_fp() retrieves the fp of a file BIO, it is a macro.

BIO_seek() is a macro that sets the position pointer to **offset** bytes from the start of file.

BIO_tell() returns the value of the position pointer.

BIO_read_filename(), **BIO_write_filename()**, **BIO_append_filename()** and **BIO_rw_filename()** set the file BIO **b** to use file **name** for reading, writing, append or read write respectively.

NOTES

When wrapping stdout, stdin or stderr the underlying stream should not normally be closed so the BIO_NOCLOSE flag should be set.

Because the file BIO calls the underlying stdio functions any quirks in stdio behaviour will be mirrored by the corresponding BIO.

On Windows BIO_new_files reserves for the filename argument to be UTF-8 encoded. In other words if you have to make it work in multi- lingual environment, encode filenames in UTF-8.

RETURN VALUES

BIO_s_file() returns the file BIO method.

BIO_new_file() and **BIO_new_fp()** return a file BIO or NULL if an error occurred.

BIO_set_fp() and **BIO_get_fp()** return 1 for success or <=0 for failure (although the current implementation never return 0).

BIO_seek() returns 0 for success or negative values for failure.

BIO_tell() returns the current file position or negative values for failure.

BIO_read_filename(), **BIO_write_filename()**, **BIO_append_filename()** and **BIO_rw_filename()** return 1 for success or <=0 for failure.

EXAMPLES

File BIO "hello world":

```
BIO *bio_out;

bio_out = BIO_new_fp(stdout, BIO_NOCLOSE);
BIO_printf(bio_out, "Hello World\n");
```

Alternative technique:

```
BIO *bio_out;

bio_out = BIO_new(BIO_s_file());
if (bio_out == NULL)
    /* Error */
if (BIO_set_fp(bio_out, stdout, BIO_NOCLOSE) <= 0)
    /* Error */
BIO_printf(bio_out, "Hello World\n");
```

Write to a file:

```
BIO *out;

out = BIO_new_file("filename.txt", "w");
if (!out)
    /* Error */
BIO_printf(out, "Hello World\n");
BIO_free(out);
```

Alternative technique:

```
BIO *out;

out = BIO_new(BIO_s_file());
if (out == NULL)
    /* Error */
if (BIO_write_filename(out, "filename.txt") <= 0)
    /* Error */
BIO_printf(out, "Hello World\n");
BIO_free(out);
```

BUGS

BIO_reset() and **BIO_seek()** are implemented using **fseek()** on the underlying stream. The return value for **fseek()** is 0 for success or -1 if an error occurred this differs from other types of BIO which will typically return 1 for success and a non positive value if an error occurred.

SEE ALSO

BIO_seek(3), **BIO_tell**(3), **BIO_reset**(3), **BIO_flush**(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

Copyright 2000-2021 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.