

NAME

COMP_CTX_new, COMP_CTX_get_method, COMP_CTX_get_type, COMP_get_type, COMP_get_name, COMP_CTX_free, COMP_compress_block, COMP_expand_block, COMP_zlib, COMP_zlib_oneshot, COMP_brotli, COMP_brotli_oneshot, COMP_zstd, COMP_zstd_oneshot, BIO_f_zlib, BIO_f_brotli, BIO_f_zstd – Compression support

SYNOPSIS

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

```
COMP_CTX *COMP_CTX_new(COMP_METHOD *meth);
void COMP_CTX_free(COMP_CTX *ctx);
const COMP_METHOD *COMP_CTX_get_method(const COMP_CTX *ctx);
int COMP_CTX_get_type(const COMP_CTX* comp);
int COMP_get_type(const COMP_METHOD *meth);
const char *COMP_get_name(const COMP_METHOD *meth);

int COMP_compress_block(COMP_CTX *ctx, unsigned char *out, int olen,
                        unsigned char *in, int ilen);
int COMP_expand_block(COMP_CTX *ctx, unsigned char *out, int olen,
                      unsigned char *in, int ilen);

COMP_METHOD *COMP_zlib(void);
COMP_METHOD *COMP_zlib_oneshot(void);
COMP_METHOD *COMP_brotli(void);
COMP_METHOD *COMP_brotli_oneshot(void);
COMP_METHOD *COMP_zstd(void);
COMP_METHOD *COMP_zstd_oneshot(void);

const BIO_METHOD *BIO_f_zlib(void);
const BIO_METHOD *BIO_f_brotli(void);
const BIO_METHOD *BIO_f_zstd(void);
```

DESCRIPTION

These functions provide compression support for OpenSSL. Compression is used within the OpenSSL library to support TLS record and certificate compression.

COMP_CTX_new() is used to create a new **COMP_CTX** structure used to compress data.

COMP_CTX_free() is used to free the returned **COMP_CTX**. If the argument is NULL, nothing is done.

COMP_CTX_get_method() returns the **COMP_METHOD** of the given *ctx*.

COMP_CTX_get_type() and **COMP_get_type()** return the NID for the **COMP_CTX** and **COMP_METHOD**, respectively. **COMP_get_name()** returns the name of the algorithm of the given **COMP_METHOD**.

COMP_compress_block() compresses *b<ilen>* bytes from the buffer *in* into the buffer *b<out>* of size *olen* using the algorithm specified by *ctx*.

COMP_expand_block() expands *ilen* bytes from the buffer *in* into the buffer *out* of size *olen* using the algorithm specified by *ctx*.

Methods (**COMP_METHOD**) may be specified by one of these functions. These functions will be available even if their corresponding compression algorithm is not configured into the OpenSSL library. In such a case, NULL will be returned.

- **COMP_zlib()** returns a **COMP_METHOD** for stream-based ZLIB compression.
- **COMP_zlib_oneshot()** returns a **COMP_METHOD** for one-shot ZLIB compression.

- **COMP_brotli()** returns a **COMP_METHOD** for stream-based Brotli compression.
- **COMP_brotli_oneshot()** returns a **COMP_METHOD** for one-shot Brotli compression.
- **COMP_zstd()** returns a **COMP_METHOD** for stream-based Zstandard compression.
- **COMP_zstd_oneshot()** returns a **COMP_METHOD** for one-shot Zstandard compression.

BIO_f_zlib(), **BIO_f_brotli()** **BIO_f_zstd()** each return a **BIO_METHOD** that may be used to create a **BIO** via **BIO_new(3)** to read and write compressed files or streams. The functions are only available if the corresponding algorithm is compiled into the OpenSSL library. NULL may be returned if the algorithm fails to load dynamically.

NOTES

While compressing non-compressible data, the output may be larger than the input. Care should be taken to size output buffers appropriate for both compression and expansion.

Compression support and compression algorithms must be enabled and built into the library before use. Refer to the INSTALL.md file when configuring OpenSSL.

ZLIB may be found at <<https://zlib.net>>

Brotli may be found at <<https://github.com/google/brotli>>.

Zstandard may be found at <<https://github.com/facebook/zstd>>.

Compression of SSL/TLS records is not recommended, as it has been shown to lead to the CRIME attack <<https://en.wikipedia.org/wiki/CRIME>>. It is disabled by default, and may be enabled by clearing the **SSL_OP_NO_COMPRESSION** option and setting the security level as appropriate. See the documentation for the **SSL_CTX_set_options(3)** and **SSL_set_options(3)** functions.

Compression is also used to support certificate compression as described in RFC8879 <<https://datatracker.ietf.org/doc/html/rfc8879>>. It may be disabled via the **SSL_OP_NO_TX_CERTIFICATE_COMPRESSION** and **SSL_OP_NO_RX_CERTIFICATE_COMPRESSION** options of the **SSL_CTX_set_options(3)** or **SSL_set_options(3)** functions.

COMP_zlib(), **COMP_brotli()** and **COMP_zstd()** are stream-based compression methods. Internal state (including compression dictionary) is maintained between calls. If an error is returned, the stream is corrupted, and should be closed.

COMP_zlib_oneshot(), **COMP_brotli_oneshot()** and **COMP_zstd_oneshot()** are not stream-based. These methods do not maintain state between calls. An error in one call does not affect future calls.

RETURN VALUES

COMP_CTX_new() returns a **COMP_CTX** on success, or NULL on failure.

COMP_CTX_get_method(), **COMP_zlib()**, **COMP_zlib_oneshot()**, **COMP_brotli()**, **COMP_brotli_oneshot()**, **COMP_zstd()**, and **COMP_zstd_oneshot()** return a **COMP_METHOD** on success, or NULL on failure.

COMP_CTX_get_type() and **COMP_get_type()** return a NID value. On failure, NID_undef is returned.

COMP_compress_block() and **COMP_expand_block()** return the number of bytes stored in the output buffer *out*. This may be 0. On failure, -1 is returned.

COMP_get_name() returns a **const char *** that must not be freed on success, or NULL on failure.

BIO_f_zlib(), **BIO_f_brotli()** and **BIO_f_zstd()** return NULL on error, and a **BIO_METHOD** on success.

SEE ALSO

BIO_new(3), **SSL_CTX_set_options(3)**, **SSL_set_options(3)**

HISTORY

Brotli and Zstandard functions were added in OpenSSL 3.2.

COPYRIGHT

Copyright 2022–2024 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>](https://www.openssl.org/source/license.html).