



Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'ERR_get_error.3ossl'

\$ man ERR_get_error.3ossl

ERR_GET_ERROR(3ossl) OpenSSL ERR_GET_ERROR(3ossl)

NAME

ERR_get_error, ERR_peek_error, ERR_peek_last_error, ERR_get_error_line,
ERR_peek_error_line, ERR_peek_last_error_line, ERR_peek_error_func,
ERR_peek_last_error_func, ERR_peek_error_data,
ERR_peek_last_error_data, ERR_get_error_all, ERR_peek_error_all,
ERR_peek_last_error_all, ERR_get_error_line_data,
ERR_peek_error_line_data, ERR_peek_last_error_line_data - obtain error
code and data

SYNOPSIS

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

unsigned long ERR_get_error(void);

unsigned long ERR_peek_error(void);

unsigned long ERR_peek_last_error(void);

unsigned long ERR_peek_error_line(const char **file, int *line);

unsigned long ERR_peek_last_error_line(const char **file, int *line);

unsigned long ERR_peek_error_func(const char **func);

unsigned long ERR_peek_last_error_func(const char **func);
```

```

unsigned long ERR_peek_error_data(const char **data, int *flags);
unsigned long ERR_peek_last_error_data(const char **data, int *flags);
unsigned long ERR_get_error_all(const char **file, int *line,
                               const char **func,
                               const char **data, int *flags);
unsigned long ERR_peek_error_all(const char **file, int *line,
                                 const char **func,
                                 const char **data, int *flags);
unsigned long ERR_peek_last_error_all(const char **file, int *line,
                                      const char *func,
                                      const char **data, int *flags);

```

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining `OPENSSL_API_COMPAT` with a suitable version value, see `openssl_user_macros(7)`:

```

unsigned long ERR_get_error_line(const char **file, int *line);
unsigned long ERR_get_error_line_data(const char **file, int *line,
                                      const char **data, int *flags);
unsigned long ERR_peek_error_line_data(const char **file, int *line,
                                       const char **data, int *flags);
unsigned long ERR_peek_last_error_line_data(const char **file, int *line,
                                            const char **data, int *flags);

```

DESCRIPTION

`ERR_get_error()` returns the earliest error code from the thread's error queue and removes the entry. This function can be called repeatedly until there are no more error codes to return.

`ERR_peek_error()` returns the earliest error code from the thread's error queue without modifying it.

`ERR_peek_last_error()` returns the latest error code from the thread's error queue without modifying it.

See `ERR_GET_LIB(3)` for obtaining further specific information such as the reason of the error, and `ERR_error_string(3)` for human-readable error messages.

`ERR_get_error_all()` is the same as `ERR_get_error()`, but on success it

additionally stores the filename, line number and function where the error occurred in `*file`, `*line` and `*func`, and also extra text and flags in `*data`, `*flags`. If any of those parameters are NULL, it will not be changed. An unset filename is indicated as "", i.e. an empty string. An unset line number is indicated as 0. An unset function name is indicated as "", i.e. an empty string.

A pointer returned this way by these functions and the ones below is valid until the respective entry is overwritten in the error queue.

`ERR_peek_error_line()` and `ERR_peek_last_error_line()` are the same as `ERR_peek_error()` and `ERR_peek_last_error()`, but on success they additionally store the filename and line number where the error occurred in `*file` and `*line`, as far as they are not NULL. An unset filename is indicated as "", i.e., an empty string. An unset line number is indicated as 0.

`ERR_peek_error_func()` and `ERR_peek_last_error_func()` are the same as `ERR_peek_error()` and `ERR_peek_last_error()`, but on success they additionally store the name of the function where the error occurred in `*func`, unless it is NULL. An unset function name is indicated as "".

`ERR_peek_error_data()` and `ERR_peek_last_error_data()` are the same as `ERR_peek_error()` and `ERR_peek_last_error()`, but on success they additionally store additional data and flags associated with the error code in `*data` and `*flags`, as far as they are not NULL. Unset data is indicated as "". In this case the value given for the flag is irrelevant (and equals 0). `*data` contains a string if `*flags&ERR_TXT_STRING` is true.

`ERR_peek_error_all()` and `ERR_peek_last_error_all()` are combinations of all of the above.

`ERR_get_error_line()`, `ERR_get_error_line_data()`, `ERR_peek_error_line_data()` and `ERR_peek_last_error_line_data()` are older variants of `ERR_peek_error_all()`, `ERR_peek_error_all()` and `ERR_peek_last_error_all()`, and may give confusing results. They should no longer be used and are therefore deprecated.

An application MUST NOT free the `*data` pointer (or any other pointers

returned by these functions) with `OPENSSL_free()` as freeing is handled automatically by the error library.

RETURN VALUES

The error code, or 0 if there is no error in the queue.

SEE ALSO

`ERR_error_string(3)`, `ERR_GET_LIB(3)`

HISTORY

`ERR_peek_error_func()`, `ERR_peek_last_error_func()`,
`ERR_peek_error_data()`, `ERR_peek_last_error_data()`, `ERR_peek_error_all()`
and `ERR_peek_last_error_all()` were added in OpenSSL 3.0.
`ERR_get_error_line()`, `ERR_get_error_line_data()`,
`ERR_peek_error_line_data()` and `ERR_peek_last_error_line_data()` became deprecated in OpenSSL 3.0.

COPYRIGHT

Copyright 2000-2022 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>>.

3.0.7 2023-07-13 ERR_GET_ERROR(3ossl)