

## NAME

PEM\_X509\_INFO\_read\_ex, PEM\_X509\_INFO\_read, PEM\_X509\_INFO\_read\_bio\_ex,  
 PEM\_X509\_INFO\_read\_bio - read PEM-encoded data structures into one or more X509\_INFO objects

## SYNOPSIS

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

```
STACK_OF(X509_INFO) *PEM_X509_INFO_read_ex(FILE *fp, STACK_OF(X509_INFO) *sk,
                                             pem_password_cb *cb, void *u,
                                             OSSL_LIB_CTX *libctx,
                                             const char *propq);
STACK_OF(X509_INFO) *PEM_X509_INFO_read(FILE *fp, STACK_OF(X509_INFO) *sk,
                                           pem_password_cb *cb, void *u);
STACK_OF(X509_INFO) *PEM_X509_INFO_read_bio_ex(BIO *bio,
                                                  STACK_OF(X509_INFO) *sk,
                                                  pem_password_cb *cb, void *u,
                                                  OSSL_LIB_CTX *libctx,
                                                  const char *propq);
STACK_OF(X509_INFO) *PEM_X509_INFO_read_bio(BIO *bp, STACK_OF(X509_INFO) *sk,
                                              pem_password_cb *cb, void *u);
```

## DESCRIPTION

**PEM\_X509\_INFO\_read\_ex()** loads the **X509\_INFO** objects from a file *fp*.

**PEM\_X509\_INFO\_read()** is similar to **PEM\_X509\_INFO\_read\_ex()** but uses the default (NULL) library context *libctx* and empty property query *propq*.

**PEM\_X509\_INFO\_read\_bio\_ex()** loads the **X509\_INFO** objects using a bio *bp*.

**PEM\_X509\_INFO\_read\_bio()** is similar to **PEM\_X509\_INFO\_read\_bio\_ex()** but uses the default (NULL) library context *libctx* and empty property query *propq*.

Each of the loaded **X509\_INFO** objects can contain a CRL, a certificate, and/or a private key. The elements are read sequentially, and as far as they are of different type than the elements read before, they are combined into the same **X509\_INFO** object. The idea behind this is that if, for instance, a certificate is followed by a private key, the private key is supposed to correspond to the certificate.

If the input stack *sk* is NULL a new stack is allocated, else the given stack is extended.

The optional *cb* and *u* parameters can be used for providing a pass phrase needed for decrypting

encrypted PEM structures (normally only private keys). See **PEM\_read\_bio\_PrivateKey(3)** and **passphrase-encoding(7)** for details.

The library context *libctx* and property query *propq* are used for fetching algorithms from providers.

## RETURN VALUES

**PEM\_X509\_INFO\_read\_ex()**, **PEM\_X509\_INFO\_read()**, **PEM\_X509\_INFO\_read\_bio\_ex()** and **PEM\_X509\_INFO\_read\_bio()** return a stack of **X509\_INFO** objects or NULL on failure.

## SEE ALSO

**PEM\_read\_bio\_ex(3)**, **PEM\_read\_bio\_PrivateKey(3)**, **passphrase-encoding(7)**

## HISTORY

The functions **PEM\_X509\_INFO\_read\_ex()** and **PEM\_X509\_INFO\_read\_bio\_ex()** were added in OpenSSL 3.0.

## COPYRIGHT

Copyright 2020-2022 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>.