



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

**at Enterprise Linux Release 9.2 Manual Pages on 'PEM\_read\_bio\_NETSCAPE\_CERT\_SEQUENCE.3oss1' con**

**`$ man PEM_read_bio_NETSCAPE_CERT_SEQUENCE.3oss1`**

PEM\_READ\_CMS(3oss1)            OpenSSL            PEM\_READ\_CMS(3oss1)

#### NAME

DECLARE\_PEM\_rw, PEM\_read\_CMS, PEM\_read\_bio\_CMS, PEM\_write\_CMS,  
PEM\_write\_bio\_CMS, PEM\_write\_DHxparams, PEM\_write\_bio\_DHxparams,  
PEM\_read\_ECCKParameters, PEM\_read\_bio\_ECCKParameters,  
PEM\_write\_ECCKParameters, PEM\_write\_bio\_ECCKParameters,  
PEM\_read\_ECPrivateKey, PEM\_write\_ECPrivateKey,  
PEM\_write\_bio\_ECPrivateKey, PEM\_read\_EC\_PUBKEY, PEM\_read\_bio\_EC\_PUBKEY,  
PEM\_write\_EC\_PUBKEY, PEM\_write\_bio\_EC\_PUBKEY,  
PEM\_read\_NETSCAPE\_CERT\_SEQUENCE, PEM\_read\_bio\_NETSCAPE\_CERT\_SEQUENCE,  
PEM\_write\_NETSCAPE\_CERT\_SEQUENCE, PEM\_write\_bio\_NETSCAPE\_CERT\_SEQUENCE,  
PEM\_read\_PKCS8, PEM\_read\_bio\_PKCS8, PEM\_write\_PKCS8,  
PEM\_write\_bio\_PKCS8, PEM\_write\_PKCS8\_PRIV\_KEY\_INFO,  
PEM\_read\_bio\_PKCS8\_PRIV\_KEY\_INFO, PEM\_read\_PKCS8\_PRIV\_KEY\_INFO,  
PEM\_write\_bio\_PKCS8\_PRIV\_KEY\_INFO, PEM\_read\_SSL\_SESSION,  
PEM\_read\_bio\_SSL\_SESSION, PEM\_write\_SSL\_SESSION,  
PEM\_write\_bio\_SSL\_SESSION, PEM\_read\_X509\_PUBKEY,  
PEM\_read\_bio\_X509\_PUBKEY, PEM\_write\_X509\_PUBKEY,  
PEM\_write\_bio\_X509\_PUBKEY - PEM object encoding routines

#### SYNOPSIS

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

```
DECLARE_PEM_rw(name, TYPE)
```

```
TYPE *PEM_read_TYPE(FILE *fp, TYPE **a, pem_password_cb *cb, void *u);
```

```
TYPE *PEM_read_bio_TYPE(BIO *bp, TYPE **a, pem_password_cb *cb, void *u);
```

```
int PEM_write_TYPE(FILE *fp, const TYPE *a);
```

```
int PEM_write_bio_TYPE(BIO *bp, const TYPE *a);
```

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)`:

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

```
int PEM_write_DHxparams(FILE *out, const DH *dh);
```

```
int PEM_write_bio_DHxparams(BIO *out, const DH *dh);
```

```
EC_GROUP *PEM_read_ECPKParameters(FILE *fp, EC_GROUP **x, pem_password_cb *cb, void *u);
```

```
EC_GROUP *PEM_read_bio_ECPKParameters(BIO *bp, EC_GROUP **x, pem_password_cb *cb, void *u);
```

```
int PEM_write_ECPKParameters(FILE *out, const EC_GROUP *x);
```

```
int PEM_write_bio_ECPKParameters(BIO *out, const EC_GROUP *x),
```

```
EC_KEY *PEM_read_EC_PUBKEY(FILE *fp, EC_KEY **x, pem_password_cb *cb, void *u);
```

```
EC_KEY *PEM_read_bio_EC_PUBKEY(BIO *bp, EC_KEY **x, pem_password_cb *cb, void *u);
```

```
int PEM_write_EC_PUBKEY(FILE *out, const EC_KEY *x);
```

```
int PEM_write_bio_EC_PUBKEY(BIO *out, const EC_KEY *x);
```

```
EC_KEY *PEM_read_ECPrivateKey(FILE *out, EC_KEY **x, pem_password_cb *cb, void *u);
```

```
EC_KEY *PEM_read_bio_ECPrivateKey(BIO *out, EC_KEY **x, pem_password_cb *cb, void *u);
```

```
int PEM_write_ECPrivateKey(FILE *out, const EC_KEY *x, const EVP_CIPHER *enc,
```

```
    const unsigned char *kstr, int klen,
```

```
    pem_password_cb *cb, void *u);
```

```
int PEM_write_bio_ECPrivateKey(BIO *out, const EC_KEY *x, const EVP_CIPHER *enc,
```

```
    const unsigned char *kstr, int klen,
```

```
pem_password_cb *cb, void *u);
```

## DESCRIPTION

All of the functions described on this page are deprecated.

Applications should use `OSSL_ENCODER_to_bio()` and `OSSL_DECODER_from_bio()` instead.

In the description below, `TYPE` is used as a placeholder for any of the OpenSSL datatypes, such as `X509`. The macro `DECLARE_PEM_rw` expands to the set of declarations shown in the next four lines of the synopsis.

These routines convert between local instances of ASN1 datatypes and the PEM encoding. For more information on the templates, see `ASN1_ITEM(3)`. For more information on the lower-level routines used by the functions here, see `PEM_read(3)`.

`PEM_read_TYPE()` reads a PEM-encoded object of `TYPE` from the file `fp` and returns it. The `cb` and `u` parameters are as described in `pem_password_cb(3)`.

`PEM_read_bio_TYPE()` is similar to `PEM_read_TYPE()` but reads from the BIO `bp`.

`PEM_write_TYPE()` writes the PEM encoding of the object `a` to the file `fp`.

`PEM_write_bio_TYPE()` similarly writes to the BIO `bp`.

## NOTES

These functions make no assumption regarding the pass phrase received from the password callback. It will simply be treated as a byte sequence.

## RETURN VALUES

PEM\_read\_TYPE() and PEM\_read\_bio\_TYPE() return a pointer to an allocated object, which should be released by calling TYPE\_free(), or NULL on error.

PEM\_write\_TYPE() and PEM\_write\_bio\_TYPE() return the number of bytes written or zero on error.

## SEE ALSO

PEM\_read(3), passphrase-encoding(7)

## HISTORY

The functions PEM\_write\_DHparams(), PEM\_write\_bio\_DHparams(), PEM\_read\_ECParameters(), PEM\_read\_bio\_ECParameters(), PEM\_write\_ECParameters(), PEM\_write\_bio\_ECParameters(), PEM\_read\_EC\_PUBKEY(), PEM\_read\_bio\_EC\_PUBKEY(), PEM\_write\_EC\_PUBKEY(), PEM\_write\_bio\_EC\_PUBKEY(), PEM\_read\_ECPrivateKey(), PEM\_read\_bio\_ECPrivateKey(), PEM\_write\_ECPrivateKey() and PEM\_write\_bio\_ECPrivateKey() were deprecated in OpenSSL 3.0.

## COPYRIGHT

Copyright 1998-2021 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>.