## NAME

d2i_PKCS8PrivateKey_bio, d2i_PKCS8PrivateKey_fp, i2d_PKCS8PrivateKey_bio, i2d_PKCS8PrivateKey_fp, i2d_PKCS8PrivateKey_nid_bio, i2d_PKCS8PrivateKey_nid_fp − PKCS#8 format private key functions

## SYNOPSIS

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

EVP_PKEY *d2i_PKCS8PrivateKey_bio(BIO *bp, EVP_PKEY **x, pem_password_cb *cb, vo
EVP_PKEY *d2i_PKCS8PrivateKey_fp(FILE *fp, EVP_PKEY **x, pem_password_cb *cb, vo

int i2d_PKCS8PrivateKey_bio(BIO *bp, EVP_PKEY *x, const EVP_CIPHER *enc,
                            char *kstr, int klen,
                            pem_password_cb *cb, void *u);

int i2d_PKCS8PrivateKey_fp(FILE *fp, EVP_PKEY *x, const EVP_CIPHER *enc,
                           char *kstr, int klen,
                           pem_password_cb *cb, void *u);

int i2d_PKCS8PrivateKey_nid_bio(BIO *bp, EVP_PKEY *x, int nid,
                                char *kstr, int klen,
                                pem_password_cb *cb, void *u);

int i2d_PKCS8PrivateKey_nid_fp(FILE *fp, EVP_PKEY *x, int nid,
                               char *kstr, int klen,
                               pem_password_cb *cb, void *u);
```

## DESCRIPTION

The PKCS#8 functions encode and decode private keys in PKCS#8 format using both PKCS#5 v1.5 and PKCS#5 v2.0 password based encryption algorithms.

Other than the use of DER as opposed to PEM these functions are identical to the corresponding **PEM** function as described in the *pem* (3) manual page.

## NOTES

Before using these functions *OpenSSL_add_all_algorithms* (3) should be called to initialize the internal algorithm lookup tables otherwise errors about unknown algorithms will occur if an attempt is made to decrypt a private key.

These functions are currently the only way to store encrypted private keys using DER format.

Currently all the functions use BIOs or FILE pointers, there are no functions which work directly on memory: this can be readily worked around by converting the buffers to memory BIOs, see *BIO_s_mem* (3) for details.

## SEE ALSO

*pem* (3)