

NAME

PKCS12_SAFEBAG_create_cert, PKCS12_SAFEBAG_create_crl,
 PKCS12_SAFEBAG_create_secret, PKCS12_SAFEBAG_create0_p8inf,
 PKCS12_SAFEBAG_create0_pkcs8, PKCS12_SAFEBAG_create_pkcs8_encrypt,
 PKCS12_SAFEBAG_create_pkcs8_encrypt_ex - Create PKCS#12 safeBag objects

SYNOPSIS

```
#include <openssl/pkcs12.h>
```

```
PKCS12_SAFEBAG *PKCS12_SAFEBAG_create_cert(X509 *x509);
PKCS12_SAFEBAG *PKCS12_SAFEBAG_create_crl(X509_CRL *crl);
PKCS12_SAFEBAG *PKCS12_SAFEBAG_create_secret(int type, int vtype,
        const unsigned char* value,
        int len);
PKCS12_SAFEBAG *PKCS12_SAFEBAG_create0_p8inf(PKCS8_PRIV_KEY_INFO *p8);
PKCS12_SAFEBAG *PKCS12_SAFEBAG_create0_pkcs8(X509_SIG *p8);
PKCS12_SAFEBAG *PKCS12_SAFEBAG_create_pkcs8_encrypt(int pbe_nid,
        const char *pass,
        int passlen,
        unsigned char *salt,
        int saltlen, int iter,
        PKCS8_PRIV_KEY_INFO *p8inf);
PKCS12_SAFEBAG *PKCS12_SAFEBAG_create_pkcs8_encrypt_ex(int pbe_nid,
        const char *pass,
        int passlen,
        unsigned char *salt,
        int saltlen, int iter,
        PKCS8_PRIV_KEY_INFO *p8inf,
        OSSL_LIB_CTX *ctx,
        const char *propq);
```

DESCRIPTION

PKCS12_SAFEBAG_create_cert() creates a new **PKCS12_SAFEBAG** of type **NID_certBag** containing the supplied certificate.

PKCS12_SAFEBAG_create_crl() creates a new **PKCS12_SAFEBAG** of type **NID_crlBag** containing the supplied crl.

PKCS12_SAFEBAG_create_secret() creates a new **PKCS12_SAFEBAG** of type corresponding to a PKCS#12 **secretBag**. The **secretBag** contents are tagged as *type* with an ASN1 value of type *vtype*

constructed using the bytes in *value* of length *len*.

PKCS12_SAFEBAG_create0_p8inf() creates a new **PKCS12_SAFEBAG** of type **NID_keyBag** containing the supplied PKCS8 structure.

PKCS12_SAFEBAG_create0_pkcs8() creates a new **PKCS12_SAFEBAG** of type **NID_pkcs8ShroudedKeyBag** containing the supplied PKCS8 structure.

PKCS12_SAFEBAG_create_pkcs8_encrypt() creates a new **PKCS12_SAFEBAG** of type **NID_pkcs8ShroudedKeyBag** by encrypting the supplied PKCS8 *p8inf*. If *pbe_nid* is 0, a default encryption algorithm is used. *pass* is the passphrase and *iter* is the iteration count. If *iter* is zero then a default value of 2048 is used. If *salt* is NULL then a salt is generated randomly.

PKCS12_SAFEBAG_create_pkcs8_encrypt_ex() is identical to **PKCS12_SAFEBAG_create_pkcs8_encrypt()** but allows for a library context *ctx* and property query *propq* to be used to select algorithm implementations.

NOTES

PKCS12_SAFEBAG_create_pkcs8_encrypt() makes assumptions regarding the encoding of the given pass phrase. See **passphrase-encoding(7)** for more information.

PKCS12_SAFEBAG_create_secret() was added in OpenSSL 3.0.

RETURN VALUES

All of these functions return a valid **PKCS12_SAFEBAG** structure or NULL if an error occurred.

CONFORMING TO

IETF RFC 7292 (<<https://tools.ietf.org/html/rfc7292>>)

SEE ALSO

PKCS12_create(3), **PKCS12_add_safe(3)**, **PKCS12_add_safes(3)**

HISTORY

PKCS12_SAFEBAG_create_pkcs8_encrypt_ex() was added in OpenSSL 3.0.

COPYRIGHT

Copyright 2019-2021 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

PKCS12_SAFEBAG_CREATE_CERT(3openssl) OpenSSL PKCS12_SAFEBAG_CREATE_CERT(3openssl)

at <<https://www.openssl.org/source/license.html>>.