NAME

PKCS12_create, PKCS12_create_ex - create a PKCS#12 structure

SYNOPSIS

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

 PKCS12 *PKCS12_create(const char *pass, const char *name, EVP_PKEY *pkey,
                       X509 *cert, STACK_OF(X509) *ca,
                       int nid_key, int nid_cert, int iter, int mac_iter, int keytype);
 PKCS12 *PKCS12_create_ex(const char *pass, const char *name, EVP_PKEY *pkey,
                          X509 *cert, STACK_OF(X509) *ca, int nid_key, int nid_cert,
                          int iter, int mac_iter, int keytype,
                          OSSL_LIB_CTX *ctx, const char *propq);
```

DESCRIPTION

PKCS12_create() creates a PKCS#12 structure.

pass is the passphrase to use. *name* is the **friendlyName** to use for the supplied certificate and key. *pkey* is the private key to include in the structure and *cert* its corresponding certificates. *ca*, if not **NULL** is an optional set of certificates to also include in the structure.

nid_key and *nid_cert* are the encryption algorithms that should be used for the key and certificate respectively. The modes GCM, CCM, XTS, and OCB are unsupported. *iter* is the encryption algorithm iteration count to use and *mac_iter* is the MAC iteration count to use. *keytype* is the type of key.

PKCS12_create_ex() is identical to **PKCS12_create()** but allows for a library context *ctx* and property query *propq* to be used to select algorithm implementations.

NOTES

The parameters *nid_key*, *nid_cert*, *iter*, *mac_iter* and *keytype* can all be set to zero and sensible defaults will be used.

These defaults are: AES password based encryption (PBES2 with PBKDF2 and AES-256-CBC) for private keys and certificates, the PBKDF2 and MAC key derivation iteration count of **PKCS12_DEFAULT_ITER** (currently 2048), and MAC algorithm HMAC with SHA2-256. The MAC key derivation algorithm used for the outer PKCS#12 structure is PKCS12KDF.

The default MAC iteration count is 1 in order to retain compatibility with old software which did not interpret MAC iteration counts. If such compatibility is not required then *mac_iter* should be set to PKCS12_DEFAULT_ITER.

keytype adds a flag to the store private key. This is a non standard extension that is only currently interpreted by MSIE. If set to zero the flag is omitted, if set to **KEY_SIG** the key can be used for signing only, if set to **KEY_EX** it can be used for signing and encryption. This option was useful for old export grade software which could use signing only keys of arbitrary size but had restrictions on the permissible sizes of keys which could be used for encryption.

If a certificate contains an *alias* or *keyid* then this will be used for the corresponding **friendlyName** or **localKeyID** in the PKCS12 structure.

Either *pkey*, *cert* or both can be **NULL** to indicate that no key or certificate is required. In previous versions both had to be present or a fatal error is returned.

nid_key or *nid_cert* can be set to -1 indicating that no encryption should be used.

mac_iter can be set to -1 and the MAC will then be omitted entirely. This can be useful when running with the FIPS provider as the PKCS12KDF is not a FIPS approvable algorithm.

PKCS12_create() makes assumptions regarding the encoding of the given pass phrase. See **passphrase-encoding**(7) for more information.

RETURN VALUES

PKCS12_create() returns a valid **PKCS12** structure or NULL if an error occurred.

CONFORMING TO

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

SEE ALSO

EVP_KDF-PKCS12KDF(7), **d2i_PKCS12**(3), **OSSL_PROVIDER-FIPS**(7), **passphrase-encoding**(7)

HISTORY

PKCS12_create_ex() was added in OpenSSL 3.0.

The defaults for encryption algorithms, MAC algorithm, and the MAC key derivation iteration count were changed in OpenSSL 3.0 to more modern standards.

COPYRIGHT