

**NAME**

PKCS12\_add\_safe, PKCS12\_add\_safe\_ex, PKCS12\_add\_safes, PKCS12\_add\_safes\_ex - Create and add objects to a PKCS#12 structure

**SYNOPSIS**

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

```
int PKCS12_add_safe(STACK_OF(PKCS7) **psafes, STACK_OF(PKCS12_SAFEBAG) *bags,
                   int safe_nid, int iter, const char *pass);
int PKCS12_add_safe_ex(STACK_OF(PKCS7) **psafes, STACK_OF(PKCS12_SAFEBAG) *bags,
                       int safe_nid, int iter, const char *pass,
                       OSSL_LIB_CTX *ctx, const char *propq);
```

```
PKCS12 *PKCS12_add_safes(STACK_OF(PKCS7) *safes, int p7_nid);
PKCS12 *PKCS12_add_safes_ex(STACK_OF(PKCS7) *safes, int p7_nid,
                             OSSL_LIB_CTX *ctx, const char *propq);
```

**DESCRIPTION**

**PKCS12\_add\_safe()** creates a new PKCS7 contentInfo containing the supplied **PKCS12\_SAFEBAGs** and adds this to a set of PKCS7 contentInfos. Its type depends on the value of **safe\_nid**:

- ⊕ If *safe\_nid* is -1, a plain PKCS7 *data* contentInfo is created.
- ⊕ If *safe\_nid* is a valid PBE algorithm NID, a PKCS7 **encryptedData** contentInfo is created. The algorithm uses *pass* as the passphrase and *iter* as the iteration count. If *iter* is zero then a default value for iteration count of 2048 is used.
- ⊕ If *safe\_nid* is 0, a PKCS7 **encryptedData** contentInfo is created using a default encryption algorithm, currently **NID\_pbe\_WithSHA1And3\_Key\_TripleDES\_CBC**.

**PKCS12\_add\_safe\_ex()** is identical to **PKCS12\_add\_safe()** but allows for a library context *ctx* and property query *propq* to be used to select algorithm implementations.

**PKCS12\_add\_safes()** creates a **PKCS12** structure containing the supplied set of PKCS7 contentInfos. The *safes* are enclosed first within a PKCS7 contentInfo of type *p7\_nid*. Currently the only supported type is **NID\_pkcs7\_data**.

**PKCS12\_add\_safes\_ex()** is identical to **PKCS12\_add\_safes()** but allows for a library context *ctx* and property query *propq* to be used to select algorithm implementations.

**NOTES**

**PKCS12\_add\_safe()** makes assumptions regarding the encoding of the given pass phrase. See **passphrase-encoding(7)** for more information.

**RETURN VALUES**

**PKCS12\_add\_safe()** returns a value of 1 indicating success or 0 for failure.

**PKCS12\_add\_safes()** returns a valid **PKCS12** structure or NULL if an error occurred.

**CONFORMING TO**

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

**SEE ALSO**

**PKCS12\_create(3)**

**HISTORY**

**PKCS12\_add\_safe\_ex()** and **PKCS12\_add\_safes\_ex()** were added in OpenSSL 3.0.

**COPYRIGHT**

Copyright 2020-2021 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>>.