

**NAME**

EVP\_PBE\_CipherInit, EVP\_PBE\_CipherInit\_ex, EVP\_PBE\_find, EVP\_PBE\_find\_ex,  
EVP\_PBE\_alg\_add\_type, EVP\_PBE\_alg\_add - Password based encryption routines

**SYNOPSIS**

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

```
int EVP_PBE_CipherInit(ASN1_OBJECT *pbe_obj, const char *pass, int passlen,
                      ASN1_TYPE *param, EVP_CIPHER_CTX *ctx, int en_de);
int EVP_PBE_CipherInit_ex(ASN1_OBJECT *pbe_obj, const char *pass, int passlen,
                          ASN1_TYPE *param, EVP_CIPHER_CTX *ctx, int en_de,
                          OSSL_LIB_CTX *libctx, const char *propq);

int EVP_PBE_find(int type, int pbe_nid, int *pcnid, int *pmnid,
                 EVP_PBE_KEYGEN **pkeygen);
int EVP_PBE_find_ex(int type, int pbe_nid, int *pcnid, int *pmnid,
                    EVP_PBE_KEYGEN **pkeygen, EVP_PBE_KEYGEN_EX **keygen_ex);

int EVP_PBE_alg_add_type(int pbe_type, int pbe_nid, int cipher_nid,
                          int md_nid, EVP_PBE_KEYGEN *keygen);
int EVP_PBE_alg_add(int nid, const EVP_CIPHER *cipher, const EVP_MD *md,
                    EVP_PBE_KEYGEN *keygen);
```

**DESCRIPTION****PBE operations**

**EVP\_PBE\_CipherInit()** and **EVP\_PBE\_CipherInit\_ex()** initialise an **EVP\_CIPHER\_CTX** *ctx* for encryption (*en\_de*=1) or decryption (*en\_de*=0) using the password *pass* of length *passlen*. The PBE algorithm type and parameters are extracted from an OID *pbe\_obj* and parameters *param*.

**EVP\_PBE\_CipherInit\_ex()** also allows the application to specify a library context *libctx* and property query *propq* to select appropriate algorithm implementations.

**PBE algorithm search**

**EVP\_PBE\_find()** and **EVP\_PBE\_find\_ex()** search for a matching algorithm using two parameters:

1. An algorithm type *type* which can be:

- ⊕ **EVP\_PBE\_TYPE\_OUTER** - A PBE algorithm
- ⊕ **EVP\_PBE\_TYPE\_PRF** - A pseudo-random function

⊕ `EVP_PBE_TYPE_KDF` - A key derivation function

2. A *pbe\_nid* which can represent the algorithm identifier with parameters e.g. `NID_pbeWithSHA1AndRC2_CBC` or an algorithm class e.g. `NID_pbes2`.

They return the algorithm's cipher ID *pcnid*, digest ID *pmnid* and a key generation function for the algorithm *pkeygen*. `EVP_PBE_CipherInit_ex()` also returns an extended key generation function *keygen\_ex* which takes a library context and property query.

If a NULL is supplied for any of *pcnid*, *pmnid*, *pkeygen* or *pkeygen\_ex* then this parameter is not returned.

### PBE algorithm add

`EVP_PBE_alg_add_type()` and `EVP_PBE_alg_add()` add an algorithm to the list of known algorithms. Their parameters have the same meaning as for `EVP_PBE_find()` and `EVP_PBE_find_ex()` functions.

### NOTES

The arguments *pbe\_obj* and *param* to `EVP_PBE_CipherInit()` and `EVP_PBE_CipherInit_ex()` together form an `X509_ALGOR` and can often be extracted directly from this structure.

### RETURN VALUES

Return value is 1 for success and 0 if an error occurred.

### SEE ALSO

`PKCS5_PBE_keyivgen(3)`, `PKCS12_PBE_keyivgen_ex(3)`, `PKCS5_v2_PBE_keyivgen_ex(3)`, `PKCS12_pbe_crypt_ex(3)`, `PKCS12_create_ex(3)`

### HISTORY

`EVP_PBE_CipherInit_ex()` and `EVP_PBE_find_ex()` were added in OpenSSL 3.0.

### COPYRIGHT

Copyright 2021-2022 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>.