

**NAME**

EVP\_PKEY\_derive\_init, EVP\_PKEY\_derive\_init\_ex, EVP\_PKEY\_derive\_set\_peer\_ex,  
EVP\_PKEY\_derive\_set\_peer, EVP\_PKEY\_derive - derive public key algorithm shared secret

**SYNOPSIS**

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

```
int EVP_PKEY_derive_init(EVP_PKEY_CTX *ctx);  
int EVP_PKEY_derive_init_ex(EVP_PKEY_CTX *ctx, const OSSL_PARAM params[]);  
int EVP_PKEY_derive_set_peer_ex(EVP_PKEY_CTX *ctx, EVP_PKEY *peer,  
                               int validate_peer);  
int EVP_PKEY_derive_set_peer(EVP_PKEY_CTX *ctx, EVP_PKEY *peer);  
int EVP_PKEY_derive(EVP_PKEY_CTX *ctx, unsigned char *key, size_t *keylen);
```

**DESCRIPTION**

**EVP\_PKEY\_derive\_init()** initializes a public key algorithm context *ctx* for shared secret derivation using the algorithm given when the context was created using **EVP\_PKEY\_CTX\_new(3)** or variants thereof. The algorithm is used to fetch a **EVP\_KEYEXCH** method implicitly, see "Implicit fetch" in **provider(7)** for more information about implicit fetches.

**EVP\_PKEY\_derive\_init\_ex()** is the same as **EVP\_PKEY\_derive\_init()** but additionally sets the passed parameters *params* on the context before returning.

**EVP\_PKEY\_derive\_set\_peer\_ex()** sets the peer key: this will normally be a public key. The *validate\_peer* will validate the public key if this value is non zero.

**EVP\_PKEY\_derive\_set\_peer()** is similar to **EVP\_PKEY\_derive\_set\_peer\_ex()** with *validate\_peer* set to 1.

**EVP\_PKEY\_derive()** derives a shared secret using *ctx*. If *key* is NULL then the maximum size of the output buffer is written to the *keylen* parameter. If *key* is not NULL then before the call the *keylen* parameter should contain the length of the *key* buffer, if the call is successful the shared secret is written to *key* and the amount of data written to *keylen*.

**NOTES**

After the call to **EVP\_PKEY\_derive\_init()**, algorithm specific control operations can be performed to set any appropriate parameters for the operation.

The function **EVP\_PKEY\_derive()** can be called more than once on the same context if several operations are performed using the same parameters.

## RETURN VALUES

**EVP\_PKEY\_derive\_init()** and **EVP\_PKEY\_derive()** return 1 for success and 0 or a negative value for failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

## EXAMPLES

Derive shared secret (for example DH or EC keys):

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

EVP_PKEY_CTX *ctx;
ENGINE *eng;
unsigned char *skey;
size_t skeylen;
EVP_PKEY *pkey, *peerkey;
/* NB: assumes pkey, eng, peerkey have been already set up */

ctx = EVP_PKEY_CTX_new(pkey, eng);
if (!ctx)
    /* Error occurred */
if (EVP_PKEY_derive_init(ctx) <= 0)
    /* Error */
if (EVP_PKEY_derive_set_peer(ctx, peerkey) <= 0)
    /* Error */

/* Determine buffer length */
if (EVP_PKEY_derive(ctx, NULL, &skeylen) <= 0)
    /* Error */

skey = OPENSSL_malloc(skeylen);

if (!skey)
    /* malloc failure */

if (EVP_PKEY_derive(ctx, skey, &skeylen) <= 0)
    /* Error */

/* Shared secret is skey bytes written to buffer skey */
```

**SEE ALSO**

**EVP\_PKEY\_CTX\_new(3), EVP\_PKEY\_encrypt(3), EVP\_PKEY\_decrypt(3), EVP\_PKEY\_sign(3),  
EVP\_PKEY\_verify(3), EVP\_PKEY\_verify\_recover(3), EVP\_KEYEXCH\_fetch(3)**

**HISTORY**

The **EVP\_PKEY\_derive\_init()**, **EVP\_PKEY\_derive\_set\_peer()** and **EVP\_PKEY\_derive()** functions were originally added in OpenSSL 1.0.0.

The **EVP\_PKEY\_derive\_init\_ex()** and **EVP\_PKEY\_derive\_set\_peer\_ex()** functions were added in OpenSSL 3.0.

**COPYRIGHT**

Copyright 2006-2023 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>.