NAME

EVP_PKEY_CTX_new, EVP_PKEY_CTX_new_id, EVP_PKEY_CTX_new_from_name, EVP_PKEY_CTX_new_from_pkey, EVP_PKEY_CTX_dup, EVP_PKEY_CTX_free, EVP_PKEY_CTX_is_a - public key algorithm context functions

SYNOPSIS

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

EVP_PKEY_CTX *EVP_PKEY_CTX_new(EVP_PKEY *pkey, ENGINE *e);
EVP_PKEY_CTX *EVP_PKEY_CTX_new_id(int id, ENGINE *e);
EVP_PKEY_CTX *EVP_PKEY_CTX_new_from_name(OSSL_LIB_CTX *libctx,
                                         const char *name,
                                         const char *propquery);
EVP_PKEY_CTX *EVP_PKEY_CTX_new_from_pkey(OSSL_LIB_CTX *libctx,
                                         EVP_PKEY *pkey,
                                         const char *propquery);
EVP_PKEY_CTX *EVP_PKEY_CTX_dup(const EVP_PKEY_CTX *ctx);
void EVP_PKEY_CTX_free(EVP_PKEY_CTX *ctx);
int EVP_PKEY_CTX_is_a(EVP_PKEY_CTX *ctx, const char *keytype);
```

DESCRIPTION

The **EVP_PKEY_CTX_new()** function allocates public key algorithm context using the *pkey* key type and ENGINE *e*.

The **EVP_PKEY_CTX_new_id()** function allocates public key algorithm context using the key type specified by *id* and ENGINE *e*.

The **EVP_PKEY_CTX_new_from_name()** function allocates a public key algorithm context using the library context *libctx* (see **OSSL_LIB_CTX**(3)), the key type specified by *name* and the property query *propquery*. None of the arguments are duplicated, so they must remain unchanged for the lifetime of the returned **EVP_PKEY_CTX** or of any of its duplicates. Read further about the possible names in "NOTES" below.

The **EVP_PKEY_CTX_new_from_pkey()** function allocates a public key algorithm context using the library context *libctx* (see **OSSL_LIB_CTX**(3)) and the algorithm specified by *pkey* and the property query *propquery*. None of the arguments are duplicated, so they must remain unchanged for the lifetime of the returned **EVP_PKEY_CTX** or any of its duplicates.

EVP_PKEY_CTX_new_id() and **EVP_PKEY_CTX_new_from_name()** are normally used when no **EVP_PKEY** structure is associated with the operations, for example during parameter generation or

key generation for some algorithms.

EVP_PKEY_CTX_dup() duplicates the context *ctx*. It is not supported for a keygen operation.

EVP_PKEY_CTX_free() frees up the context *ctx*. If *ctx* is NULL, nothing is done.

EVP_PKEY_is_a() checks if the key type associated with *ctx* is *keytype*.

NOTES

On EVP_PKEY_CTX

The **EVP_PKEY_CTX** structure is an opaque public key algorithm context used by the OpenSSL high-level public key API. Contexts **MUST NOT** be shared between threads: that is it is not permissible to use the same context simultaneously in two threads.

On Key Types

We mention "key type" in this manual, which is the same as "algorithm" in most cases, allowing either term to be used interchangeably. There are algorithms where the *key type* and the *algorithm* of the operations that use the keys are not the same, such as EC keys being used for ECDSA and ECDH operations.

Key types are given in two different manners:

Legacy NID or EVP_PKEY type

This is the *id* used with **EVP_PKEY_CTX_new_id()**.

These are **EVP_PKEY_RSA**, **EVP_PKEY_RSA_PSS**, **EVP_PKEY_DSA**, **EVP_PKEY_DH**, **EVP_PKEY_EC**, **EVP_PKEY_SM2**, **EVP_PKEY_X25519**, **EVP_PKEY_X448**, and are used by legacy methods.

Name strings

This is the *name* used with **EVP_PKEY_CTX_new_from_name()**.

These are names like "RSA", "DSA", and what's available depends on what providers are currently accessible.

The OpenSSL providers offer a set of key types available this way, please see **OSSL_PROVIDER-FIPS**(7) and **OSSL_PROVIDER-default**(7) and related documentation for more information.

RETURN VALUES

EVP_PKEY_CTX_new(), **EVP_PKEY_CTX_new_id()** and **EVP_PKEY_CTX_dup()** return either the newly allocated **EVP_PKEY_CTX** structure or **NULL** if an error occurred.

EVP_PKEY_CTX_free() does not return a value.

EVP_PKEY_CTX_is_a() returns 1 for true and 0 for false.

SEE ALSO

EVP_PKEY_new(3)

HISTORY

The **EVP_PKEY_CTX_new()**, **EVP_PKEY_CTX_new_id()**, **EVP_PKEY_CTX_dup()** and **EVP_PKEY_CTX_free()** functions were added in OpenSSL 1.0.0.

The **EVP_PKEY_CTX_new_from_name()** and **EVP_PKEY_CTX_new_from_pkey()** functions were added in OpenSSL 3.0.

COPYRIGHT