Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'EVP_PKEY_can_sign.3ossl' command

$ man EVP_PKEY_can_sign.3ossl

EVP_PKEY_IS_A(3ossl) OpenSSL EVP_PKEY_IS_A(3ossl)

NAME

EVP_PKEY_is_a, EVP_PKEY_can_sign, EVP_PKEY_type_names_do_all,

EVP_PKEY_get0_type_name, EVP_PKEY_get0_description,

EVP_PKEY_get0_provider - key type and capabilities functions

SYNOPSIS

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

int EVP_PKEY_is_a(const EVP_PKEY *pkey, const char *name);

int EVP_PKEY_can_sign(const EVP_PKEY *pkey);

int EVP_PKEY_type_names_do_all(const EVP_PKEY *pkey,
                               void (*fn)(const char *name, void *data),
                               void *data);

const char *EVP_PKEY_get0_type_name(const EVP_PKEY *key);

const char *EVP_PKEY_get0_description(const EVP_PKEY *key);

const OSSL_PROVIDER *EVP_PKEY_get0_provider(const EVP_PKEY *key);
```

DESCRIPTION

EVP_PKEY_is_a() checks if the key type of pkey is name.

EVP_PKEY_can_sign() checks if the functionality for the key type of pkey supports signing. No other check is done, such as whether pkey contains a private key.

EVP_PKEY_type_names_do_all() traverses all names for pkey's key type, and calls fn with each name and data. For example, an RSA EVP_PKEY may be named both "RSA" and "rsaEncryption". The order of the names

depends on the provider implementation that holds the key.

EVP_PKEY_get0_type_name() returns the first key type name that is found for the given pkey. Note that the pkey may have multiple synonyms associated with it. In this case it depends on the provider implementation that holds the key which one will be returned.

Ownership of the returned string is retained by the pkey object and should not be freed by the caller.

EVP_PKEY_get0_description() returns a description of the type of EVP_PKEY, meant for display and human consumption. The description is at the discretion of the key type implementation.

EVP_PKEY_get0_provider() returns the provider of the EVP_PKEY's EVP_KEYMGMT(3).

RETURN VALUES

EVP_PKEY_is_a() returns 1 if pkey has the key type name, otherwise 0.

EVP_PKEY_can_sign() returns 1 if the pkey key type functionality supports signing, otherwise 0.

EVP_PKEY_get0_type_name() returns the name that is found or NULL on error.

EVP_PKEY_get0_description() returns the description if found or NULL if not.

EVP_PKEY_get0_provider() returns the provider if found or NULL if not.

EVP_PKEY_type_names_do_all() returns 1 if the callback was called for all names. A return value of 0 means that the callback was not called for any names.

EXAMPLES

EVP_PKEY_is_a()

The loaded providers and what key types they support will ultimately determine what name is possible to use with EVP_PKEY_is_a(). We do know that the default provider supports RSA, DH, DSA and EC keys, so we can use this as an crude example:

```
#include <openssl/evp.h>
...
    /* |pkey| is an EVP_PKEY* */
```

```
if (EVP_PKEY_is_a(pkey, "RSA")) {
    BIGNUM *modulus = NULL;
    if (EVP_PKEY_get_bn_param(pkey, "n", &modulus))
        /* do whatever with the modulus */
    BN_free(modulus);
}
```

EVP_PKEY_can_sign()

```
#include <openssl/evp.h>
...
    /* |pkey| is an EVP_PKEY* */
    if (!EVP_PKEY_can_sign(pkey)) {
        fprintf(stderr, "Not a signing key!");
        exit(1);
    }
    /* Sign something... */
```

HISTORY

The functions described here were added in OpenSSL 3.0.

COPYRIGHT