

**NAME**

EVP\_PKEY\_gettable\_params, EVP\_PKEY\_get\_params, EVP\_PKEY\_get\_int\_param,  
 EVP\_PKEY\_get\_size\_t\_param, EVP\_PKEY\_get\_bn\_param, EVP\_PKEY\_get\_utf8\_string\_param,  
 EVP\_PKEY\_get\_octet\_string\_param - retrieve key parameters from a key

**SYNOPSIS**

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

```
const OSSL_PARAM *EVP_PKEY_gettable_params(EVP_PKEY *pkey);
int EVP_PKEY_get_params(const EVP_PKEY *pkey, OSSL_PARAM params[]);
int EVP_PKEY_get_int_param(const EVP_PKEY *pkey, const char *key_name,
                           int *out);
int EVP_PKEY_get_size_t_param(const EVP_PKEY *pkey, const char *key_name,
                              size_t *out);
int EVP_PKEY_get_bn_param(const EVP_PKEY *pkey, const char *key_name,
                          BIGNUM **bn);
int EVP_PKEY_get_utf8_string_param(const EVP_PKEY *pkey, const char *key_name,
                                   char *str, size_t max_buf_sz,
                                   size_t *out_len);
int EVP_PKEY_get_octet_string_param(const EVP_PKEY *pkey, const char *key_name,
                                    unsigned char *buf, size_t max_buf_sz,
                                    size_t *out_len);
```

**DESCRIPTION**

See **OSSL\_PARAM(3)** for information about parameters.

**EVP\_PKEY\_get\_params()** retrieves parameters from the key *pkey*, according to the contents of *params*.

**EVP\_PKEY\_gettable\_params()** returns a constant list of *params* indicating the names and types of key parameters that can be retrieved.

An **OSSL\_PARAM(3)** of type **OSSL\_PARAM\_INTEGER** or **OSSL\_PARAM\_UNSIGNED\_INTEGER** is of arbitrary length. Such a parameter can be obtained using any of the functions **EVP\_PKEY\_get\_int\_param()**, **EVP\_PKEY\_get\_size\_t\_param()** or **EVP\_PKEY\_get\_bn\_param()**. Attempting to obtain an integer value that does not fit into a native C **int** type will cause **EVP\_PKEY\_get\_int\_param()** to fail. Similarly attempting to obtain an integer value that is negative or does not fit into a native C **size\_t** type using **EVP\_PKEY\_get\_size\_t\_param()** will also fail.

**EVP\_PKEY\_get\_int\_param()** retrieves a key *pkey* integer value *\*out* associated with a name of *key\_name* if it fits into "int" type. For parameters that do not fit into "int" use **EVP\_PKEY\_get\_bn\_param()**.

**EVP\_PKEY\_get\_size\_t\_param()** retrieves a key *pkey* size\_t value *\*out* associated with a name of *key\_name* if it fits into "size\_t" type. For parameters that do not fit into "size\_t" use **EVP\_PKEY\_get\_bn\_param()**.

**EVP\_PKEY\_get\_bn\_param()** retrieves a key *pkey* BIGNUM value *\*\*bn* associated with a name of *key\_name*. If *\*bn* is NULL then the BIGNUM is allocated by the method.

**EVP\_PKEY\_get\_utf8\_string\_param()** get a key *pkey* UTF8 string value into a buffer *str* of maximum size *max\_buf\_sz* associated with a name of *key\_name*. The maximum size must be large enough to accommodate the string value including a terminating NUL byte, or this function will fail. If *out\_len* is not NULL, *\*out\_len* is set to the length of the string not including the terminating NUL byte. The required buffer size not including the terminating NUL byte can be obtained from *\*out\_len* by calling the function with *str* set to NULL.

**EVP\_PKEY\_get\_octet\_string\_param()** get a key *pkey*'s octet string value into a buffer *buf* of maximum size *max\_buf\_sz* associated with a name of *key\_name*. If *out\_len* is not NULL, *\*out\_len* is set to the length of the contents. The required buffer size can be obtained from *\*out\_len* by calling the function with *buf* set to NULL.

## NOTES

These functions only work for **EVP\_PKEY**s that contain a provider side key.

## RETURN VALUES

**EVP\_PKEY\_gettable\_params()** returns NULL on error or if it is not supported.

All other methods return 1 if a value associated with the key's *key\_name* was successfully returned, or 0 if there was an error. An error may be returned by methods **EVP\_PKEY\_get\_utf8\_string\_param()** and **EVP\_PKEY\_get\_octet\_string\_param()** if *max\_buf\_sz* is not big enough to hold the value. If *out\_len* is not NULL, *\*out\_len* will be assigned the required buffer size to hold the value.

## EXAMPLES

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

char curve_name[64];
unsigned char pub[256];
BIGNUM *bn_priv = NULL;
```

```

/*
 * NB: assumes 'key' is set up before the next step. In this example the key
 * is an EC key.
 */

if (!EVP_PKEY_get_utf8_string_param(key, OSSL_PKEY_PARAM_GROUP_NAME,
                                     curve_name, sizeof(curve_name), &len)) {
    /* Error */
}
if (!EVP_PKEY_get_octet_string_param(key, OSSL_PKEY_PARAM_PUB_KEY,
                                      pub, sizeof(pub), &len)) {
    /* Error */
}
if (!EVP_PKEY_get_bn_param(key, OSSL_PKEY_PARAM_PRIV_KEY, &bn_priv)) {
    /* Error */
}

BN_clear_free(bn_priv);

```

**SEE ALSO**

**EVP\_PKEY\_CTX\_new(3)**, **provider-keymgmt(7)**, **OSSL\_PARAM(3)**

**HISTORY**

These functions were added in OpenSSL 3.0.

**COPYRIGHT**

Copyright 2020-2023 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>.