



*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 'X509\_PUBKEY\_get.3ossl' command**

**\$ man X509\_PUBKEY\_get.3ossl**

X509\_PUBKEY\_NEW(3ossl)      OpenSSL      X509\_PUBKEY\_NEW(3ossl)

### NAME

X509\_PUBKEY\_new\_ex, X509\_PUBKEY\_new, X509\_PUBKEY\_free, X509\_PUBKEY\_dup,  
X509\_PUBKEY\_set, X509\_PUBKEY\_get0, X509\_PUBKEY\_get, d2i\_PUBKEY\_ex,  
d2i\_PUBKEY, i2d\_PUBKEY, d2i\_PUBKEY\_bio, d2i\_PUBKEY\_fp, i2d\_PUBKEY\_fp,  
i2d\_PUBKEY\_bio, X509\_PUBKEY\_set0\_param, X509\_PUBKEY\_get0\_param,  
X509\_PUBKEY\_eq - SubjectPublicKeyInfo public key functions

### SYNOPSIS

```
#include <openssl/x509.h>

X509_PUBKEY *X509_PUBKEY_new_ex(OSSL_LIB_CTX *libctx, const char *propq);
X509_PUBKEY *X509_PUBKEY_new(void);
void X509_PUBKEY_free(X509_PUBKEY *a);
X509_PUBKEY *X509_PUBKEY_dup(const X509_PUBKEY *a);
int X509_PUBKEY_set(X509_PUBKEY **x, EVP_PKEY *pkey);
EVP_PKEY *X509_PUBKEY_get0(const X509_PUBKEY *key);
EVP_PKEY *X509_PUBKEY_get(const X509_PUBKEY *key);
EVP_PKEY *d2i_PUBKEY_ex(EVP_PKEY **a, const unsigned char **pp, long length,
                        OSSL_LIB_CTX *libctx, const char *propq);
EVP_PKEY *d2i_PUBKEY(EVP_PKEY **a, const unsigned char **pp, long length);
int i2d_PUBKEY(const EVP_PKEY *a, unsigned char **pp);
EVP_PKEY *d2i_PUBKEY_bio(BIO *bp, EVP_PKEY **a);
EVP_PKEY *d2i_PUBKEY_fp(FILE *fp, EVP_PKEY **a);
int i2d_PUBKEY_fp(const FILE *fp, EVP_PKEY *pkey);
```

```

int i2d_PUBKEY_bio(BIO *bp, const EVP_PKEY *pkey);
int X509_PUBKEY_set0_param(X509_PUBKEY *pub, ASN1_OBJECT *aobj,
    int ptype, void *pval,
    unsigned char *penc, int penclen);
int X509_PUBKEY_get0_param(ASN1_OBJECT **ppkalg,
    const unsigned char **pk, int *ppklen,
    X509_ALGOR **pa, const X509_PUBKEY *pub);
int X509_PUBKEY_eq(X509_PUBKEY *a, X509_PUBKEY *b);

```

## DESCRIPTION

The X509\_PUBKEY structure represents the ASN.1 SubjectPublicKeyInfo structure defined in RFC5280 and used in certificates and certificate requests.

X509\_PUBKEY\_new\_ex() allocates and initializes an X509\_PUBKEY structure associated with the given OSSL\_LIB\_CTX in the libctx parameter. Any algorithm fetches associated with using the X509\_PUBKEY object will use the property query string propq. See "ALGORITHM FETCHING" in crypto(7) for further information about algorithm fetching.

X509\_PUBKEY\_new() is the same as X509\_PUBKEY\_new\_ex() except that the default (NULL) OSSL\_LIB\_CTX and a NULL property query string are used.

X509\_PUBKEY\_dup() creates a duplicate copy of the X509\_PUBKEY object specified by a.

X509\_PUBKEY\_free() frees up X509\_PUBKEY structure a. If a is NULL nothing is done.

X509\_PUBKEY\_set() sets the public key in \*x to the public key contained in the EVP\_PKEY structure pkey. If \*x is not NULL any existing public key structure will be freed.

X509\_PUBKEY\_get0() returns the public key contained in key. The returned value is an internal pointer which MUST NOT be freed after use.

X509\_PUBKEY\_get() is similar to X509\_PUBKEY\_get0() except the reference count on the returned key is incremented so it MUST be freed using EVP\_PKEY\_free() after use.

d2i\_PUBKEY\_ex() decodes an EVP\_PKEY structure using

SubjectPublicKeyInfo format. Some public key decoding implementations may use cryptographic algorithms. In this case the supplied library context libctx and property query string propq are used. d2i\_PUBKEY() does the same as d2i\_PUBKEY\_ex() except that the default library context and property query string are used.

i2d\_PUBKEY() encodes an EVP\_PKEY structure using SubjectPublicKeyInfo format.

d2i\_PUBKEY\_bio(), d2i\_PUBKEY\_fp(), i2d\_PUBKEY\_bio() and i2d\_PUBKEY\_fp() are similar to d2i\_PUBKEY() and i2d\_PUBKEY() except they decode or encode using a BIO or FILE pointer.

X509\_PUBKEY\_set0\_param() sets the public key parameters of pub. The OID associated with the algorithm is set to aobj. The type of the algorithm parameters is set to type using the structure pval. The encoding of the public key itself is set to the penclen bytes contained in buffer penc. On success ownership of all the supplied parameters is passed to pub so they must not be freed after the call.

X509\_PUBKEY\_get0\_param() retrieves the public key parameters from pub, \*ppkalg is set to the associated OID and the encoding consists of \*ppklen bytes at \*pk, \*pa is set to the associated AlgorithmIdentifier for the public key. If the value of any of these parameters is not required it can be set to NULL. All of the retrieved pointers are internal and must not be freed after the call.

X509\_PUBKEY\_eq() compares two X509\_PUBKEY values.

## NOTES

The X509\_PUBKEY functions can be used to encode and decode public keys in a standard format.

In many cases applications will not call the X509\_PUBKEY functions directly: they will instead call wrapper functions such as X509\_get0\_pubkey().

## RETURN VALUES

If the allocation fails, X509\_PUBKEY\_new() and X509\_PUBKEY\_dup() return NULL and set an error code that can be obtained by ERR\_get\_error(3).

Otherwise they return a pointer to the newly allocated structure.

X509\_PUBKEY\_free() does not return a value.

X509\_PUBKEY\_get0() and X509\_PUBKEY\_get() return a pointer to an EVP\_PKEY structure or NULL if an error occurs.

X509\_PUBKEY\_set(), X509\_PUBKEY\_set0\_param() and X509\_PUBKEY\_get0\_param() return 1 for success and 0 if an error occurred.

X509\_PUBKEY\_eq() returns 1 for equal, 0 for different, and < 0 on error.

#### SEE ALSO

d2i\_X509(3), ERR\_get\_error(3), X509\_get\_pubkey(3),

#### HISTORY

The X509\_PUBKEY\_new\_ex() and X509\_PUBKEY\_eq() functions were added in OpenSSL 3.0.

#### COPYRIGHT

Copyright 2016-2021 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>>.

3.0.7                    2023-07-13            X509\_PUBKEY\_NEW(3ossl)