NAME

DH_get0_pqg, DH_set0_pqg, DH_get0_key, DH_set0_key, DH_get0_p, DH_get0_q, DH_get0_g, DH_get0_priv_key, DH_get0_pub_key, DH_clear_flags, DH_test_flags, DH_set_flags, DH_get0_engine, DH_get_length, DH_set_length − Routines for getting and setting data in a DH object

SYNOPSIS

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

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining **OPENSSL_API_COMPAT** with a suitable version value, see **openssl_user_macros**(7):

```
void DH_get0_pqg(const DH *dh,
                 const BIGNUM **p, const BIGNUM **q, const BIGNUM **g);
int DH_set0_pqg(DH *dh, BIGNUM *p, BIGNUM *q, BIGNUM *g);
void DH_get0_key(const DH *dh,
                 const BIGNUM **pub_key, const BIGNUM **priv_key);
int DH_set0_key(DH *dh, BIGNUM *pub_key, BIGNUM *priv_key);
const BIGNUM *DH_get0_p(const DH *dh);
const BIGNUM *DH_get0_q(const DH *dh);
const BIGNUM *DH_get0_g(const DH *dh);
const BIGNUM *DH_get0_priv_key(const DH *dh);
const BIGNUM *DH_get0_pub_key(const DH *dh);
void DH_clear_flags(DH *dh, int flags);
int DH_test_flags(const DH *dh, int flags);
void DH_set_flags(DH *dh, int flags);

long DH_get_length(const DH *dh);
int DH_set_length(DH *dh, long length);

ENGINE *DH_get0_engine(DH *d);
```

DESCRIPTION

All of the functions described on this page are deprecated. Applications should instead use **EVP_PKEY_get_bn_param**(3) for any methods that return a **BIGNUM**. Refer to **EVP_PKEY−DH**(7) for more information.

A DH object contains the parameters *p*, *q* and *g*. Note that the *q* parameter is optional. It also contains a public key (*pub_key*) and (optionally) a private key (*priv_key*).

The *p*, *q* and *g* parameters can be obtained by calling **DH_get0_pqg()**. If the parameters have not yet been set then **p*, **q* and **g* will be set to NULL. Otherwise they are set to pointers to their respective values. These point directly to the internal representations of the values and therefore should not be freed directly. Any of the out parameters *p*, *q*, and *g* can be NULL, in which case no value will be returned for that parameter.

The *p*, *q* and *g* values can be set by calling **DH_set0_pqg()** and passing the new values for *p*, *q* and *g* as parameters to the function. Calling this function transfers the memory management of the values to the DH object, and therefore the values that have been passed in should not be freed directly after this function has been called. The *q* parameter may be NULL. **DH_set0_pqg()** also checks if the parameters associated with *p* and *g* and optionally *q* are associated with known safe prime groups. If it is a safe prime group then the value of *q* will be set to q = (p − 1) / 2 if *q* is NULL. The optional length parameter will be set to BN_num_bits(*q*) if *q* is not NULL.

To get the public and private key values use the **DH_get0_key()** function. A pointer to the public key will be stored in **pub_key*, and a pointer to the private key will be stored in **priv_key*. Either may be NULL if they have not been set yet, although if the private key has been set then the public key must be. The values point to the internal representation of the public key and private key values. This memory should not be freed directly. Any of the out parameters *pub_key* and *priv_key* can be NULL, in which case no value will be returned for that parameter.

The public and private key values can be set using **DH_set0_key()**. Either parameter may be NULL, which means the corresponding DH field is left untouched. As with **DH_set0_pqg()** this function transfers the memory management of the key values to the DH object, and therefore they should not be freed directly after this function has been called.

Any of the values *p*, *q*, *g*, *priv_key*, and *pub_key* can also be retrieved separately by the corresponding function **DH_get0_p()**, **DH_get0_q()**, **DH_get0_g()**, **DH_get0_priv_key()**, and **DH_get0_pub_key()**, respectively.

DH_set_flags() sets the flags in the *flags* parameter on the DH object. Multiple flags can be passed in one go (bitwise ORed together). Any flags that are already set are left set. **DH_test_flags()** tests to see whether the flags passed in the *flags* parameter are currently set in the DH object. Multiple flags can be tested in one go. All flags that are currently set are returned, or zero if none of the flags are set. **DH_clear_flags()** clears the specified flags within the DH object.

DH_get0_engine() returns a handle to the ENGINE that has been set for this DH object, or NULL if no such ENGINE has been set. This function is deprecated. All engines should be replaced by providers.

The **DH_get_length()** and **DH_set_length()** functions get and set the optional length parameter associated with this DH object. If the length is nonzero then it is used, otherwise it is ignored. The *length* parameter indicates the length of the secret exponent (private key) in bits. For safe prime groups the optional length parameter *length* can be set to a value greater or equal to 2 * maximum_target_security_strength(BN_num_bits(*p*)) as listed in SP800−56Ar3 Table(s) 25 & 26. These functions are deprecated and should be replaced with **EVP_PKEY_CTX_set_params()** and **EVP_PKEY_get_int_param()** using the parameter key **OSSL_PKEY_PARAM_DH_PRIV_LEN** as described in **EVP_PKEY−DH**(7).

NOTES

Values retrieved with **DH_get0_key()** are owned by the DH object used in the call and may therefore *not* be passed to **DH_set0_key()**. If needed, duplicate the received value using **BN_dup()** and pass the duplicate. The same applies to **DH_get0_pqg()** and **DH_set0_pqg()**.

RETURN VALUES

DH_set0_pqg() and **DH_set0_key()** return 1 on success or 0 on failure.

DH_get0_p(), **DH_get0_q()**, **DH_get0_g()**, **DH_get0_priv_key()**, and **DH_get0_pub_key()** return the respective value, or NULL if it is unset.

DH_test_flags() returns the current state of the flags in the DH object.

DH_get0_engine() returns the ENGINE set for the DH object or NULL if no ENGINE has been set.

DH_get_length() returns the length of the secret exponent (private key) in bits, or zero if no such length has been explicitly set.

SEE ALSO

DH_new(3), **DH_new**(3), **DH_generate_parameters**(3), **DH_generate_key**(3), **DH_set_method**(3), **DH_size**(3), **DH_meth_new**(3)

HISTORY

The functions described here were added in OpenSSL 1.1.0.

All of these functions were deprecated in OpenSSL 3.0.

COPYRIGHT