NAME

DSA_get0_pqg, DSA_set0_pqg, DSA_get0_key, DSA_set0_key, DSA_get0_p, DSA_get0_q, DSA_get0_g, DSA_get0_pub_key, DSA_get0_priv_key, DSA_clear_flags, DSA_test_flags, DSA_set_flags, DSA_get0_engine - Routines for getting and setting data in a DSA object

SYNOPSIS

```
#include <openssl/dsa.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 DSA_get0_pqg(const DSA *d,
                  const BIGNUM **p, const BIGNUM **q, const BIGNUM **g);
int DSA_set0_pqg(DSA *d, BIGNUM *p, BIGNUM *q, BIGNUM *g);
void DSA_get0_key(const DSA *d,
                  const BIGNUM **pub_key, const BIGNUM **priv_key);
int DSA_set0_key(DSA *d, BIGNUM *pub_key, BIGNUM *priv_key);
const BIGNUM *DSA_get0_p(const DSA *d);
const BIGNUM *DSA_get0_q(const DSA *d);
const BIGNUM *DSA_get0_g(const DSA *d);
const BIGNUM *DSA_get0_pub_key(const DSA *d);
const BIGNUM *DSA_get0_priv_key(const DSA *d);
void DSA_clear_flags(DSA *d, int flags);
int DSA_test_flags(const DSA *d, int flags);
void DSA_set_flags(DSA *d, int flags);
ENGINE *DSA_get0_engine(DSA *d);
```

DESCRIPTION

All of the functions described on this page are deprecated. Applications should instead use **EVP_PKEY_get_bn_param**(3).

A DSA object contains the parameters **p**, **q** and **g**. 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 **DSA_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.

The **p**, **q** and **g** values can be set by calling **DSA_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 DSA object, and therefore the values that have been passed in should not be freed directly after this function has been called.

To get the public and private key values use the **DSA_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.

The public and private key values can be set using **DSA_set0_key()**. The public key must be non-NULL the first time this function is called on a given DSA object. The private key may be NULL. On subsequent calls, either may be NULL, which means the corresponding DSA field is left untouched. As for **DSA_set0_pqg()** this function transfers the memory management of the key values to the DSA 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 **DSA_get0_p()**, **DSA_get0_q()**, **DSA_get0_g()**, **DSA_get0_priv_key()**, and **DSA_get0_pub_key()**, respectively.

DSA_set_flags() sets the flags in the **flags** parameter on the DSA object. Multiple flags can be passed in one go (bitwise ORed together). Any flags that are already set are left set. **DSA_test_flags()** tests to see whether the flags passed in the **flags** parameter are currently set in the DSA 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. **DSA_clear_flags()** clears the specified flags within the DSA object.

DSA_get0_engine() returns a handle to the ENGINE that has been set for this DSA object, or NULL if no such ENGINE has been set.

NOTES

Values retrieved with **DSA_get0_key()** are owned by the DSA object used in the call and may therefore *not* be passed to **DSA_set0_key()**. If needed, duplicate the received value using **BN_dup()** and pass the duplicate. The same applies to **DSA_get0_pqg()** and **DSA_set0_pqg()**.

RETURN VALUES

DSA_set0_pqg() and **DSA_set0_key()** return 1 on success or 0 on failure.

DSA_test_flags() returns the current state of the flags in the DSA object.

DSA_get0_engine() returns the ENGINE set for the DSA object or NULL if no ENGINE has been set.

SEE ALSO

EVP_PKEY_get_bn_param(3), **DSA_new**(3), **DSA_new**(3), **DSA_generate_parameters**(3), **DSA_generate_key**(3), **DSA_dup_DH**(3), **DSA_do_sign**(3), **DSA_set_method**(3), **DSA_SIG_new**(3), **DSA_sign**(3), **DSA_size**(3), **DSA_meth_new**(3)

HISTORY

The functions described here were added in OpenSSL 1.1.0 and deprecated in OpenSSL 3.0.

COPYRIGHT