

NAME

DSA_generate_parameters_ex, DSA_generate_parameters – generate DSA parameters

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):

```
int DSA_generate_parameters_ex(DSA *dsa, int bits,
                              const unsigned char *seed, int seed_len,
                              int *counter_ret, unsigned long *h_ret,
                              BN_GENCB *cb);
```

The following functions have been deprecated since OpenSSL 0.9.8, and can be hidden entirely by defining **OPENSSL_API_COMPAT** with a suitable version value, see **openssl_user_macros** (7):

```
DSA *DSA_generate_parameters(int bits, unsigned char *seed, int seed_len,
                             int *counter_ret, unsigned long *h_ret,
                             void (*callback)(int, int, void *), void *cb_arg);
```

DESCRIPTION

All of the functions described on this page are deprecated. Applications should instead use **EVP_PKEY_paramgen_init** (3) and **EVP_PKEY_keygen** (3) as described in **EVP_PKEY-DSA** (7).

DSA_generate_parameters_ex() generates primes *p* and *q* and a generator *g* for use in the DSA and stores the result in **dsa**.

bits is the length of the prime *p* to be generated. For lengths under 2048 bits, the length of *q* is 160 bits; for lengths greater than or equal to 2048 bits, the length of *q* is set to 256 bits.

If **seed** is **NULL**, the primes will be generated at random. If **seed_len** is less than the length of *q*, an error is returned.

DSA_generate_parameters_ex() places the iteration count in ***counter_ret** and a counter used for finding a generator in ***h_ret**, unless these are **NULL**.

A callback function may be used to provide feedback about the progress of the key generation. If **cb** is not **NULL**, it will be called as shown below. For information on the **BN_GENCB** structure and the **BN_GENCB_call** function discussed below, refer to **BN_generate_prime** (3).

DSA_generate_parameters() is similar to **DSA_generate_parameters_ex()** but expects an old-style callback function; see **BN_generate_prime** (3) for information on the old-style callback.

- When a candidate for *q* is generated, **BN_GENCB_call(cb, 0, m++)** is called (*m* is 0 for the first candidate).
- When a candidate for *q* has passed a test by trial division, **BN_GENCB_call(cb, 1, -1)** is called. While a candidate for *q* is tested by Miller–Rabin primality tests, **BN_GENCB_call(cb, 1, i)** is called in the outer loop (once for each witness that confirms that the candidate may be prime); *i* is the loop counter (starting at 0).
- When a prime *q* has been found, **BN_GENCB_call(cb, 2, 0)** and **BN_GENCB_call(cb, 3, 0)** are called.
- Before a candidate for *p* (other than the first) is generated and tested, **BN_GENCB_call(cb, 0, counter)** is called.
- When a candidate for *p* has passed the test by trial division, **BN_GENCB_call(cb, 1, -1)** is called. While it is tested by the Miller–Rabin primality test, **BN_GENCB_call(cb, 1, i)** is called in the outer loop (once for each witness that confirms that the candidate may be prime). *i* is the loop counter (starting at 0).
- When *p* has been found, **BN_GENCB_call(cb, 2, 1)** is called.

- When the generator has been found, **BN_GENCB_call(cb, 3, 1)** is called.

RETURN VALUES

DSA_generate_parameters_ex() returns a 1 on success, or 0 otherwise. The error codes can be obtained by **ERR_get_error(3)**.

DSA_generate_parameters() returns a pointer to the DSA structure or **NULL** if the parameter generation fails.

BUGS

Seed lengths greater than 20 are not supported.

SEE ALSO

DSA_new(3), **ERR_get_error(3)**, **RAND_bytes(3)**, **DSA_free(3)**, **BN_generate_prime(3)**

HISTORY

DSA_generate_parameters_ex() was deprecated in OpenSSL 3.0.

DSA_generate_parameters() was deprecated in OpenSSL 0.9.8; use **DSA_generate_parameters_ex()** instead.

COPYRIGHT

Copyright 2000–2021 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>](https://www.openssl.org/source/license.html).