NAME

DSA_set_default_method, DSA_get_default_method, DSA_set_method, DSA_new_method, DSA_OpenSSL − select DSA method

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_set_default_method(const DSA_METHOD *meth);

const DSA_METHOD *DSA_get_default_method(void);

int DSA_set_method(DSA *dsa, const DSA_METHOD *meth);

DSA *DSA_new_method(ENGINE *engine);

const DSA_METHOD *DSA_OpenSSL(void);
```

DESCRIPTION

All of the functions described on this page are deprecated. Applications should providers instead of method overrides.

A **DSA_METHOD** specifies the functions that OpenSSL uses for DSA operations. By modifying the method, alternative implementations such as hardware accelerators may be used. IMPORTANT: See the NOTES section for important information about how these DSA API functions are affected by the use of **ENGINE** API calls.

Initially, the default DSA_METHOD is the OpenSSL internal implementation, as returned by **DSA_OpenSSL()**.

DSA_set_default_method() makes **meth** the default method for all DSA structures created later. **NB**: This is true only whilst no ENGINE has been set as a default for DSA, so this function is no longer recommended. This function is not thread−safe and should not be called at the same time as other OpenSSL functions.

DSA_get_default_method() returns a pointer to the current default DSA_METHOD. However, the meaningfulness of this result is dependent on whether the ENGINE API is being used, so this function is no longer recommended.

DSA_set_method() selects **meth** to perform all operations using the key **rsa**. This will replace the DSA_METHOD used by the DSA key and if the previous method was supplied by an ENGINE, the handle to that ENGINE will be released during the change. It is possible to have DSA keys that only work with certain DSA_METHOD implementations (e.g. from an ENGINE module that supports embedded hardware−protected keys), and in such cases attempting to change the DSA_METHOD for the key can have unexpected results. See **DSA_meth_new** (3) for information on constructing custom DSA_METHOD objects;

DSA_new_method() allocates and initializes a DSA structure so that **engine** will be used for the DSA operations. If **engine** is NULL, the default engine for DSA operations is used, and if no default ENGINE is set, the DSA_METHOD controlled by **DSA_set_default_method()** is used.

RETURN VALUES

DSA_OpenSSL() and **DSA_get_default_method()** return pointers to the respective **DSA_METHOD**s.

DSA_set_default_method() returns no value.

DSA_set_method() returns nonzero if the provided **meth** was successfully set as the method for **dsa** (including unloading the ENGINE handle if the previous method was supplied by an ENGINE).

DSA_new_method() returns NULL and sets an error code that can be obtained by **ERR_get_error** (3) if the allocation fails. Otherwise it returns a pointer to the newly allocated structure.

SEE ALSO

DSA_new (3), **DSA_new** (3), **DSA_meth_new** (3)

HISTORY

All of these functions were deprecated in OpenSSL 3.0.

COPYRIGHT