NAME

DH_set_default_method, DH_get_default_method, DH_set_method, DH_new_method, DH_OpenSSL - select DH method

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_set_default_method(const DH_METHOD *meth);

const DH_METHOD *DH_get_default_method(void);

int DH_set_method(DH *dh, const DH_METHOD *meth);

DH *DH_new_method(ENGINE *engine);

const DH_METHOD *DH_OpenSSL(void);
```

DESCRIPTION

All of the functions described on this page are deprecated. Applications should instead use the provider APIs.

A **DH_METHOD** specifies the functions that OpenSSL uses for Diffie-Hellman 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 DH API functions are affected by the use of **ENGINE** API calls.

Initially, the default DH_METHOD is the OpenSSL internal implementation, as returned by **DH_OpenSSL()**.

DH_set_default_method() makes **meth** the default method for all DH structures created later. **NB**: This is true only whilst no ENGINE has been set as a default for DH, 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.

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

DH_set_method() selects **meth** to perform all operations using the key **dh**. This will replace the DH_METHOD used by the DH 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 DH keys that only work with certain DH_METHOD implementations (e.g. from an ENGINE module that supports embedded hardware-protected keys), and in such cases attempting to change the DH_METHOD for the key can have unexpected results.

DH_new_method() allocates and initializes a DH structure so that **engine** will be used for the DH operations. If **engine** is NULL, the default ENGINE for DH operations is used, and if no default ENGINE is set, the DH_METHOD controlled by **DH_set_default_method()** is used.

A new DH_METHOD object may be constructed using **DH_meth_new()** (see **DH_meth_new**(3)).

RETURN VALUES

DH_OpenSSL() and **DH_get_default_method()** return pointers to the respective **DH_METHOD**s.

DH_set_default_method() returns no value.

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

DH_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

DH_new(3), **DH_new**(3), **DH_meth_new**(3)

HISTORY

All of these functions were deprecated in OpenSSL 3.0.

COPYRIGHT