

**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**

Copyright 2000–2020 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>>.