NAME

DH_meth_new, DH_meth_free, DH_meth_dup, DH_meth_get0_name, DH_meth_set1_name, DH_meth_get_flags, DH_meth_set_flags, DH_meth_get0_app_data, DH_meth_set0_app_data, DH_meth_get_generate_key, DH_meth_set_generate_key, DH_meth_get_compute_key, DH_meth_set_compute_key, DH_meth_get_bn_mod_exp, DH_meth_set_bn_mod_exp, DH_meth_get_init, DH_meth_set_init, DH_meth_get_finish, DH_meth_set_finish, DH_meth_get_generate_params, DH_meth_set_generate_params - Routines to build up DH methods

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):

```
DH_METHOD *DH_meth_new(const char *name, int flags);

void DH_meth_free(DH_METHOD *dhm);

DH_METHOD *DH_meth_dup(const DH_METHOD *dhm);

const char *DH_meth_get0_name(const DH_METHOD *dhm);
int DH_meth_set1_name(DH_METHOD *dhm, const char *name);

int DH_meth_get_flags(const DH_METHOD *dhm);
int DH_meth_set_flags(DH_METHOD *dhm, int flags);

void *DH_meth_get0_app_data(const DH_METHOD *dhm);
int DH_meth_set0_app_data(DH_METHOD *dhm, void *app_data);

int (*DH_meth_get_generate_key(const DH_METHOD *dhm))(DH *);
int DH_meth_set_generate_key(DH_METHOD *dhm, int (*generate_key)(DH *));

int (*DH_meth_get_compute_key(const DH_METHOD *dhm))
   (unsigned char *key, const BIGNUM *pub_key, DH *dh);
int DH_meth_set_compute_key(DH_METHOD *dhm,
   int (*compute_key)(unsigned char *key, const BIGNUM *pub_key, DH *dh));

int (*DH_meth_get_bn_mod_exp(const DH_METHOD *dhm))
   (const DH *dh, BIGNUM *r, const BIGNUM *a, const BIGNUM *p,
    const BIGNUM *m, BN_CTX *ctx, BN_MONT_CTX *m_ctx);
```

```
int DH_meth_set_bn_mod_exp(DH_METHOD *dhm,
   int (*bn_mod_exp)(const DH *dh, BIGNUM *r, const BIGNUM *a,
                     const BIGNUM *p, const BIGNUM *m, BN_CTX *ctx,
                     BN_MONT_CTX *m_ctx));

int (*DH_meth_get_init(const DH_METHOD *dhm))(DH *);
int DH_meth_set_init(DH_METHOD *dhm, int (*init)(DH *));

int (*DH_meth_get_finish(const DH_METHOD *dhm))(DH *);
int DH_meth_set_finish(DH_METHOD *dhm, int (*finish)(DH *));

int (*DH_meth_get_generate_params(const DH_METHOD *dhm))
   (DH *, int, int, BN_GENCB *);
int DH_meth_set_generate_params(DH_METHOD *dhm,
   int (*generate_params)(DH *, int, int, BN_GENCB *));
```

DESCRIPTION

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

The **DH_METHOD** type is a structure used for the provision of custom DH implementations. It provides a set of functions used by OpenSSL for the implementation of the various DH capabilities.

DH_meth_new() creates a new **DH_METHOD** structure. It should be given a unique **name** and a set of **flags**. The **name** should be a NULL terminated string, which will be duplicated and stored in the **DH_METHOD** object. It is the callers responsibility to free the original string. The flags will be used during the construction of a new **DH** object based on this **DH_METHOD**. Any new **DH** object will have those flags set by default.

DH_meth_dup() creates a duplicate copy of the **DH_METHOD** object passed as a parameter. This might be useful for creating a new **DH_METHOD** based on an existing one, but with some differences.

DH_meth_free() destroys a **DH_METHOD** structure and frees up any memory associated with it.

DH_meth_get0_name() will return a pointer to the name of this DH_METHOD. This is a pointer to the internal name string and so should not be freed by the caller. **DH_meth_set1_name()** sets the name of the DH_METHOD to **name**. The string is duplicated and the copy is stored in the DH_METHOD structure, so the caller remains responsible for freeing the memory associated with the name.

DH_meth_get_flags() returns the current value of the flags associated with this DH_METHOD.

DH_meth_set_flags() provides the ability to set these flags.

The functions **DH_meth_get0_app_data()** and **DH_meth_set0_app_data()** provide the ability to associate implementation specific data with the DH_METHOD. It is the application's responsibility to free this data before the DH_METHOD is freed via a call to **DH_meth_free()**.

DH_meth_get_generate_key() and **DH_meth_set_generate_key()** get and set the function used for generating a new DH key pair respectively. This function will be called in response to the application calling **DH_generate_key()**. The parameter for the function has the same meaning as for **DH_generate_key()**.

DH_meth_get_compute_key() and **DH_meth_set_compute_key()** get and set the function used for computing a new DH shared secret respectively. This function will be called in response to the application calling **DH_compute_key()**. The parameters for the function have the same meaning as for **DH_compute_key()**.

DH_meth_get_bn_mod_exp() and **DH_meth_set_bn_mod_exp()** get and set the function used for computing the following value:

r = a ^ p mod m

This function will be called by the default OpenSSL function for **DH_generate_key()**. The result is stored in the **r** parameter. This function may be NULL unless using the default generate key function, in which case it must be present.

DH_meth_get_init() and **DH_meth_set_init()** get and set the function used for creating a new DH instance respectively. This function will be called in response to the application calling **DH_new()** (if the current default DH_METHOD is this one) or **DH_new_method()**. The **DH_new()** and **DH_new_method()** functions will allocate the memory for the new DH object, and a pointer to this newly allocated structure will be passed as a parameter to the function. This function may be NULL.

DH_meth_get_finish() and **DH_meth_set_finish()** get and set the function used for destroying an instance of a DH object respectively. This function will be called in response to the application calling **DH_free()**. A pointer to the DH to be destroyed is passed as a parameter. The destroy function should be used for DH implementation specific clean up. The memory for the DH itself should not be freed by this function. This function may be NULL.

DH_meth_get_generate_params() and **DH_meth_set_generate_params()** get and set the function used for generating DH parameters respectively. This function will be called in response to the application calling **DH_generate_parameters_ex()** (or **DH_generate_parameters()**). The parameters for the function

have the same meaning as for **DH_generate_parameters_ex()**. This function may be NULL.

RETURN VALUES

DH_meth_new() and **DH_meth_dup()** return the newly allocated DH_METHOD object or NULL on failure.

DH_meth_get0_name() and **DH_meth_get_flags()** return the name and flags associated with the DH_METHOD respectively.

All other DH_meth_get_*() functions return the appropriate function pointer that has been set in the DH_METHOD, or NULL if no such pointer has yet been set.

DH_meth_set1_name() and all DH_meth_set_*() functions return 1 on success or 0 on failure.

SEE ALSO

DH_new(3), **DH_new**(3), **DH_generate_parameters**(3), **DH_generate_key**(3), **DH_set_method**(3), **DH_size**(3), **DH_get0_pqg**(3)

HISTORY

All of these functions were deprecated in OpenSSL 3.0.

The functions described here were added in OpenSSL 1.1.0.

COPYRIGHT