

NAME

X509_LOOKUP_METHOD, X509_LOOKUP_meth_new, X509_LOOKUP_meth_free,
 X509_LOOKUP_meth_set_new_item, X509_LOOKUP_meth_get_new_item,
 X509_LOOKUP_meth_set_free, X509_LOOKUP_meth_get_free, X509_LOOKUP_meth_set_init,
 X509_LOOKUP_meth_get_init, X509_LOOKUP_meth_set_shutdown,
 X509_LOOKUP_meth_get_shutdown, X509_LOOKUP_ctrl_fn, X509_LOOKUP_meth_set_ctrl,
 X509_LOOKUP_meth_get_ctrl, X509_LOOKUP_get_by_subject_fn,
 X509_LOOKUP_meth_set_get_by_subject, X509_LOOKUP_meth_get_get_by_subject,
 X509_LOOKUP_get_by_issuer_serial_fn, X509_LOOKUP_meth_set_get_by_issuer_serial,
 X509_LOOKUP_meth_get_get_by_issuer_serial, X509_LOOKUP_get_by_fingerprint_fn,
 X509_LOOKUP_meth_set_get_by_fingerprint, X509_LOOKUP_meth_get_get_by_fingerprint,
 X509_LOOKUP_get_by_alias_fn, X509_LOOKUP_meth_set_get_by_alias,
 X509_LOOKUP_meth_get_get_by_alias, X509_OBJECT_set1_X509,
 X509_OBJECT_set1_X509_CRL - Routines to build up X509_LOOKUP methods

SYNOPSIS

```
#include <openssl/x509_vfy.h>
```

```
typedef x509_lookup_method_st X509_LOOKUP_METHOD;
```

```
X509_LOOKUP_METHOD *X509_LOOKUP_meth_new(const char *name);
```

```
void X509_LOOKUP_meth_free(X509_LOOKUP_METHOD *method);
```

```
int X509_LOOKUP_meth_set_new_item(X509_LOOKUP_METHOD *method,  
    int (*new_item) (X509_LOOKUP *ctx));
```

```
int (*X509_LOOKUP_meth_get_new_item(const X509_LOOKUP_METHOD* method))  
    (X509_LOOKUP *ctx);
```

```
int X509_LOOKUP_meth_set_free(X509_LOOKUP_METHOD *method,  
    void (*free) (X509_LOOKUP *ctx));
```

```
void (*X509_LOOKUP_meth_get_free(const X509_LOOKUP_METHOD* method))  
    (X509_LOOKUP *ctx);
```

```
int X509_LOOKUP_meth_set_init(X509_LOOKUP_METHOD *method,  
    int (*init) (X509_LOOKUP *ctx));
```

```
int (*X509_LOOKUP_meth_get_init(const X509_LOOKUP_METHOD* method))  
    (X509_LOOKUP *ctx);
```

```
int X509_LOOKUP_meth_set_shutdown(X509_LOOKUP_METHOD *method,  
    int (*shutdown) (X509_LOOKUP *ctx));
```

```
int (*X509_LOOKUP_meth_get_shutdown(const X509_LOOKUP_METHOD* method))
    (X509_LOOKUP *ctx);

typedef int (*X509_LOOKUP_ctrl_fn)(X509_LOOKUP *ctx, int cmd, const char *argc,
    long argl, char **ret);
int X509_LOOKUP_meth_set_ctrl(X509_LOOKUP_METHOD *method,
    X509_LOOKUP_ctrl_fn ctrl_fn);
X509_LOOKUP_ctrl_fn X509_LOOKUP_meth_get_ctrl(const X509_LOOKUP_METHOD *method);

typedef int (*X509_LOOKUP_get_by_subject_fn)(X509_LOOKUP *ctx,
    X509_LOOKUP_TYPE type,
    const X509_NAME *name,
    X509_OBJECT *ret);
int X509_LOOKUP_meth_set_get_by_subject(X509_LOOKUP_METHOD *method,
    X509_LOOKUP_get_by_subject_fn fn);
X509_LOOKUP_get_by_subject_fn X509_LOOKUP_meth_get_get_by_subject(
    const X509_LOOKUP_METHOD *method);

typedef int (*X509_LOOKUP_get_by_issuer_serial_fn)(X509_LOOKUP *ctx,
    X509_LOOKUP_TYPE type,
    const X509_NAME *name,
    const ASN1_INTEGER *serial,
    X509_OBJECT *ret);
int X509_LOOKUP_meth_set_get_by_issuer_serial(
    X509_LOOKUP_METHOD *method, X509_LOOKUP_get_by_issuer_serial_fn fn);
X509_LOOKUP_get_by_issuer_serial_fn X509_LOOKUP_meth_get_get_by_issuer_serial(
    const X509_LOOKUP_METHOD *method);

typedef int (*X509_LOOKUP_get_by_fingerprint_fn)(X509_LOOKUP *ctx,
    X509_LOOKUP_TYPE type,
    const unsigned char* bytes,
    int len,
    X509_OBJECT *ret);
int X509_LOOKUP_meth_set_get_by_fingerprint(X509_LOOKUP_METHOD *method,
    X509_LOOKUP_get_by_fingerprint_fn fn);
X509_LOOKUP_get_by_fingerprint_fn X509_LOOKUP_meth_get_get_by_fingerprint(
    const X509_LOOKUP_METHOD *method);

typedef int (*X509_LOOKUP_get_by_alias_fn)(X509_LOOKUP *ctx,
    X509_LOOKUP_TYPE type,
```

```

        const char *str,
        int len,
        X509_OBJECT *ret);
int X509_LOOKUP_meth_set_get_by_alias(X509_LOOKUP_METHOD *method,
    X509_LOOKUP_get_by_alias_fn fn);
X509_LOOKUP_get_by_alias_fn X509_LOOKUP_meth_get_get_by_alias(
    const X509_LOOKUP_METHOD *method);

int X509_OBJECT_set1_X509(X509_OBJECT *a, X509 *obj);
int X509_OBJECT_set1_X509_CRL(X509_OBJECT *a, X509_CRL *obj);

```

DESCRIPTION

The **X509_LOOKUP_METHOD** type is a structure used for the implementation of new X509_LOOKUP types. It provides a set of functions used by OpenSSL for the implementation of various X509 and X509_CRL lookup capabilities. One instance of an X509_LOOKUP_METHOD can be associated to many instantiations of an **X509_LOOKUP** structure.

X509_LOOKUP_meth_new() creates a new **X509_LOOKUP_METHOD** structure. It should be given a human-readable string containing a brief description of the lookup method.

X509_LOOKUP_meth_free() destroys a **X509_LOOKUP_METHOD** structure.

X509_LOOKUP_get_new_item() and **X509_LOOKUP_set_new_item()** get and set the function that is called when an **X509_LOOKUP** object is created with **X509_LOOKUP_new()**. If an X509_LOOKUP_METHOD requires any per-X509_LOOKUP specific data, the supplied new_item function should allocate this data and invoke **X509_LOOKUP_set_method_data(3)**.

X509_LOOKUP_get_free() and **X509_LOOKUP_set_free()** get and set the function that is used to free any method data that was allocated and set from within new_item function.

X509_LOOKUP_meth_get_init() and **X509_LOOKUP_meth_set_init()** get and set the function that is used to initialize the method data that was set with **X509_LOOKUP_set_method_data(3)** as part of the new_item routine.

X509_LOOKUP_meth_get_shutdown() and **X509_LOOKUP_meth_set_shutdown()** get and set the function that is used to shut down the method data whose state was previously initialized in the init function.

X509_LOOKUP_meth_get_ctrl() and **X509_LOOKUP_meth_set_ctrl()** get and set a function to be used to handle arbitrary control commands issued by **X509_LOOKUP_ctrl()**. The control function is

given the `X509_LOOKUP ctx`, along with the arguments passed by `X509_LOOKUP_ctrl`. `cmd` is an arbitrary integer that defines some operation. `argc` is a pointer to an array of characters. `argl` is an integer. `ret`, if set, points to a location where any return data should be written to. How `argc` and `argl` are used depends entirely on the control function.

`X509_LOOKUP_set_get_by_subject()`, `X509_LOOKUP_set_get_by_issuer_serial()`, `X509_LOOKUP_set_get_by_fingerprint()`, `X509_LOOKUP_set_get_by_alias()` set the functions used to retrieve an X509 or X509_CRL object by the object's subject, issuer, fingerprint, and alias respectively. These functions are given the `X509_LOOKUP` context, the type of the `X509_OBJECT` being requested, parameters related to the lookup, and an `X509_OBJECT` that will receive the requested object.

Implementations must add objects they find to the `X509_STORE` object using `X509_STORE_add_cert()` or `X509_STORE_add_crl()`. This increments its reference count. However, the `X509_STORE_CTX_get_by_subject()` function also increases the reference count which leads to one too many references being held. Therefore, applications should additionally call `X509_free()` or `X509_CRL_free()` to decrement the reference count again.

Implementations should also use either `X509_OBJECT_set1_X509()` or `X509_OBJECT_set1_X509_CRL()` to set the result. Note that this also increments the result's reference count.

Any method data that was created as a result of the `new_item` function set by `X509_LOOKUP_meth_set_new_item()` can be accessed with `X509_LOOKUP_get_method_data(3)`. The `X509_STORE` object that owns the `X509_LOOKUP` may be accessed with `X509_LOOKUP_get_store(3)`. Successful lookups should return 1, and unsuccessful lookups should return 0.

`X509_LOOKUP_get_get_by_subject()`, `X509_LOOKUP_get_get_by_issuer_serial()`, `X509_LOOKUP_get_get_by_fingerprint()`, `X509_LOOKUP_get_get_by_alias()` retrieve the function set by the corresponding setter.

RETURN VALUES

The `X509_LOOKUP_meth_set` functions return 1 on success or 0 on error.

The `X509_LOOKUP_meth_get` functions return the corresponding function pointers.

SEE ALSO

`X509_STORE_new(3)`, `SSL_CTX_set_cert_store(3)`

HISTORY

The functions described here were added in OpenSSL 1.1.0i.

COPYRIGHT

Copyright 2018-2020 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>.