NAME

X509_STORE_set_lookup_crls_cb, X509_STORE_set_verify_func, X509_STORE_get_cleanup, X509_STORE_set_cleanup, X509_STORE_get_lookup_crls, X509_STORE_set_lookup_crls, X509_STORE_get_lookup_certs, X509_STORE_set_lookup_certs, X509_STORE_get_check_policy, X509_STORE_set_check_policy, X509_STORE_get_cert_crl, X509_STORE_set_cert_crl, X509_STORE_get_check_crl, X509_STORE_set_check_crl, X509_STORE_get_get_crl, X509_STORE_set_get_crl, X509_STORE_get_check_revocation, X509_STORE_set_check_revocation, X509_STORE_get_check_issued, X509_STORE_set_check_issued, X509_STORE_CTX_get1_issuer, X509_STORE_get_get_issuer, X509_STORE_set_get_issuer, X509_STORE_CTX_get_verify, X509_STORE_set_verify, X509_STORE_get_verify_cb, X509_STORE_set_verify_cb_func, X509_STORE_set_verify_cb, X509_STORE_CTX_cert_crl_fn, X509_STORE_CTX_check_crl_fn, X509_STORE_CTX_check_issued_fn, X509_STORE_CTX_check_policy_fn, X509_STORE_CTX_check_revocation_fn, X509_STORE_CTX_cleanup_fn, X509_STORE_CTX_get_crl_fn, X509_STORE_CTX_get_issuer_fn, X509_STORE_CTX_lookup_certs_fn, X509_STORE_CTX_lookup_crls_fn - set verification callback

SYNOPSIS

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

typedef int (*X509_STORE_CTX_get_issuer_fn)(X509 **issuer,
                                            X509_STORE_CTX *ctx, X509 *x);
typedef int (*X509_STORE_CTX_check_issued_fn)(X509_STORE_CTX *ctx,
                                              X509 *x, X509 *issuer);
typedef int (*X509_STORE_CTX_check_revocation_fn)(X509_STORE_CTX *ctx);
typedef int (*X509_STORE_CTX_get_crl_fn)(X509_STORE_CTX *ctx,
                                         X509_CRL **crl, X509 *x);
typedef int (*X509_STORE_CTX_check_crl_fn)(X509_STORE_CTX *ctx, X509_CRL *crl);
typedef int (*X509_STORE_CTX_cert_crl_fn)(X509_STORE_CTX *ctx,
                                          X509_CRL *crl, X509 *x);
typedef int (*X509_STORE_CTX_check_policy_fn)(X509_STORE_CTX *ctx);
typedef STACK_OF(X509) *(*X509_STORE_CTX_lookup_certs_fn)(X509_STORE_CTX *ctx,
                                                          const X509_NAME *nm);
typedef STACK_OF(X509_CRL) *(*X509_STORE_CTX_lookup_crls_fn)(const
                                                             X509_STORE_CTX *ctx,
                                                             const X509_NAME *nm);
typedef int (*X509_STORE_CTX_cleanup_fn)(X509_STORE_CTX *ctx);

void X509_STORE_set_verify_cb(X509_STORE *ctx,
                              X509_STORE_CTX_verify_cb verify_cb);
```

```
X509_STORE_CTX_verify_cb X509_STORE_get_verify_cb(const X509_STORE_CTX *ctx);

void X509_STORE_set_verify(X509_STORE *ctx, X509_STORE_CTX_verify_fn verify);
X509_STORE_CTX_verify_fn X509_STORE_CTX_get_verify(const X509_STORE_CTX *ctx);

int X509_STORE_CTX_get1_issuer(X509 **issuer, X509_STORE_CTX *ctx, X509 *x);
X509_STORE_CTX_get_issuer_fn X509_STORE_get_get_issuer(const X509_STORE_CTX *ctx);
void X509_STORE_set_get_issuer(X509_STORE *ctx,
                               X509_STORE_CTX_get_issuer_fn get_issuer);

void X509_STORE_set_check_issued(X509_STORE *ctx,
                                 X509_STORE_CTX_check_issued_fn check_issued);
X509_STORE_CTX_check_issued_fn
  X509_STORE_get_check_issued(const X509_STORE_CTX *ctx);

void X509_STORE_set_check_revocation(X509_STORE *ctx,
                                     X509_STORE_CTX_check_revocation_fn check_revocation);
X509_STORE_CTX_check_revocation_fn
  X509_STORE_get_check_revocation(const X509_STORE_CTX *ctx);

void X509_STORE_set_get_crl(X509_STORE *ctx,
                            X509_STORE_CTX_get_crl_fn get_crl);
X509_STORE_CTX_get_crl_fn X509_STORE_get_get_crl(const X509_STORE_CTX *ctx);

void X509_STORE_set_check_crl(X509_STORE *ctx,
                              X509_STORE_CTX_check_crl_fn check_crl);
X509_STORE_CTX_check_crl_fn
  X509_STORE_get_check_crl(const X509_STORE_CTX *ctx);

void X509_STORE_set_cert_crl(X509_STORE *ctx,
                             X509_STORE_CTX_cert_crl_fn cert_crl);
X509_STORE_CTX_cert_crl_fn X509_STORE_get_cert_crl(const X509_STORE_CTX *ctx);

void X509_STORE_set_check_policy(X509_STORE *ctx,
                                 X509_STORE_CTX_check_policy_fn check_policy);
X509_STORE_CTX_check_policy_fn
  X509_STORE_get_check_policy(const X509_STORE_CTX *ctx);

void X509_STORE_set_lookup_certs(X509_STORE *ctx,
                                 X509_STORE_CTX_lookup_certs_fn lookup_certs);
```

```
X509_STORE_CTX_lookup_certs_fn
   X509_STORE_get_lookup_certs(const X509_STORE_CTX *ctx);

void X509_STORE_set_lookup_crls(X509_STORE *ctx,
                        X509_STORE_CTX_lookup_crls_fn lookup_crls);
X509_STORE_CTX_lookup_crls_fn
   X509_STORE_get_lookup_crls(const X509_STORE_CTX *ctx);

void X509_STORE_set_cleanup(X509_STORE *ctx,
                    X509_STORE_CTX_cleanup_fn cleanup);
X509_STORE_CTX_cleanup_fn X509_STORE_get_cleanup(const X509_STORE_CTX *ctx);

/* Aliases */
void X509_STORE_set_verify_cb_func(X509_STORE *st,
                          X509_STORE_CTX_verify_cb verify_cb);
void X509_STORE_set_verify_func(X509_STORE *ctx,
                        X509_STORE_CTX_verify_fn verify);
void X509_STORE_set_lookup_crls_cb(X509_STORE *ctx,
                          X509_STORE_CTX_lookup_crls_fn lookup_crls);
```

DESCRIPTION

X509_STORE_set_verify_cb() sets the verification callback of *ctx* to *verify_cb* overwriting the previous callback. The callback assigned with this function becomes a default for the one that can be assigned directly to the corresponding **X509_STORE_CTX**, please see **X509_STORE_CTX_set_verify_cb**(3) for further information.

X509_STORE_set_verify() sets the final chain verification function for *ctx* to *verify*. Its purpose is to go through the chain of certificates and check that all signatures are valid and that the current time is within the limits of each certificate's first and last validity time. The final chain verification functions must return 0 on failure and 1 on success. *If no chain verification function is provided, the internal default function will be used instead.*

X509_STORE_CTX_get1_issuer() tries to find a certificate from the *store* component of *ctx* with a subject name matching the issuer name of *x*. On success it assigns to **issuer* the first match that is currently valid, or at least the most recently expired match if there is no currently valid one. If the function returns 1 the caller is responsible for freeing **issuer*.

X509_STORE_set_get_issuer() sets the function *get_issuer* to get the "best" candidate issuer certificate of the given certificate *x*. When such a certificate is found, *get_issuer* must up-ref and assign it to **issuer* and then return 1. Otherwise *get_issuer* must return 0 if not found and -1 (or 0) on failure. If

X509_STORE_set_get_issuer() is not used or *get_issuer* is NULL then **X509_STORE_CTX_get1_issuer()** is used as the default implementation.

X509_STORE_set_check_issued() sets the function to check that a given certificate *x* is issued by the issuer certificate *issuer*. This function must return 0 on failure (among others if *x* hasn't been issued with *issuer*) and 1 on success. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_set_check_revocation() sets the revocation checking function. Its purpose is to look through the final chain and check the revocation status for each certificate. It must return 0 on failure and 1 on success. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_set_get_crl() sets the function to get the crl for a given certificate *x*. When found, the crl must be assigned to **crl*. This function must return 0 on failure and 1 on success. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_set_check_crl() sets the function to check the validity of the given *crl*. This function must return 0 on failure and 1 on success. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_set_cert_crl() sets the function to check the revocation status of the given certificate *x* against the given *crl*. This function must return 0 on failure and 1 on success. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_set_check_policy() sets the function to check the policies of all the certificates in the final chain.. This function must return 0 on failure and 1 on success. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_set_lookup_certs() and **X509_STORE_set_lookup_crls()** set the functions to look up all the certs or all the CRLs that match the given name *nm*. These functions return NULL on failure and a pointer to a stack of certificates (**X509**) or to a stack of CRLs (**X509_CRL**) on success. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_set_cleanup() sets the final cleanup function, which is called when the context (**X509_STORE_CTX**) is being torn down. This function doesn't return any value. *If no function to get the issuer is provided, the internal default function will be used instead.*

X509_STORE_get_verify_cb(), **X509_STORE_CTX_get_verify()**, **X509_STORE_get_get_issuer()**, **X509_STORE_get_check_issued()**, **X509_STORE_get_check_revocation()**,

X509_STORE_get_get_crl(), **X509_STORE_get_check_crl()**, **X509_STORE_set_verify()**, **X509_STORE_set_get_issuer()**, **X509_STORE_get_cert_crl()**, **X509_STORE_get_check_policy()**, **X509_STORE_get_lookup_certs()**, **X509_STORE_get_lookup_crls()** and **X509_STORE_get_cleanup()** all return the function pointer assigned with **X509_STORE_set_check_issued()**, **X509_STORE_set_check_revocation()**, **X509_STORE_set_get_crl()**, **X509_STORE_set_check_crl()**, **X509_STORE_set_cert_crl()**, **X509_STORE_set_check_policy()**, **X509_STORE_set_lookup_certs()**, **X509_STORE_set_lookup_crls()** and **X509_STORE_set_cleanup()**, or NULL if no assignment has been made.

X509_STORE_set_verify_cb_func(), **X509_STORE_set_verify_func()** and **X509_STORE_set_lookup_crls_cb()** are aliases for **X509_STORE_set_verify_cb()**, **X509_STORE_set_verify()** and X509_STORE_set_lookup_crls, available as macros for backward compatibility.

NOTES

All the callbacks from a **X509_STORE** are inherited by the corresponding **X509_STORE_CTX** structure when it is initialized. See **X509_STORE_CTX_set_verify_cb**(3) for further details.

BUGS

The macro version of this function was the only one available before OpenSSL 1.0.0.

RETURN VALUES

The X509_STORE_set_*() functions do not return a value.

The X509_STORE_get_*() functions return a pointer of the appropriate function type.

X509_STORE_CTX_get1_issuer() returns 1 if a suitable certificate is found, 0 if not found, -1 on other error.

SEE ALSO

X509_STORE_CTX_set_verify_cb(3), **X509_STORE_CTX_get0_chain**(3), **X509_STORE_CTX_verify_cb**(3), **X509_STORE_CTX_verify_fn**(3), **CMS_verify**(3)

HISTORY

The **X509_STORE_set_verify_cb()** function was added in OpenSSL 1.0.0.

The functions **X509_STORE_set_verify_cb()**, **X509_STORE_get_verify_cb()**, **X509_STORE_set_verify()**, **X509_STORE_CTX_get_verify()**, **X509_STORE_set_get_issuer()**, **X509_STORE_get_get_issuer()**, **X509_STORE_set_check_issued()**,

X509_STORE_get_check_issued(), **X509_STORE_set_check_revocation()**, **X509_STORE_get_check_revocation()**, **X509_STORE_set_get_crl()**, **X509_STORE_get_get_crl()**, **X509_STORE_set_check_crl()**, **X509_STORE_get_check_crl()**, **X509_STORE_set_cert_crl()**, **X509_STORE_get_cert_crl()**, **X509_STORE_set_check_policy()**, **X509_STORE_get_check_policy()**, **X509_STORE_set_lookup_certs()**, **X509_STORE_get_lookup_certs()**, **X509_STORE_set_lookup_crls()**, **X509_STORE_get_lookup_crls()**, **X509_STORE_set_cleanup()** and **X509_STORE_get_cleanup()** were added in OpenSSL 1.1.0.

COPYRIGHT