## NAME

OSSL_SELF_TEST_new, OSSL_SELF_TEST_free, OSSL_SELF_TEST_onbegin, OSSL_SELF_TEST_oncorrupt_byte, OSSL_SELF_TEST_onend - functionality to trigger a callback during a self test

## SYNOPSIS

#include <openssl/self_test.h>

 OSSL_SELF_TEST *OSSL_SELF_TEST_new(OSSL_CALLBACK *cb, void *cbarg);
 void OSSL_SELF_TEST_free(OSSL_SELF_TEST *st);

 void OSSL_SELF_TEST_onbegin(OSSL_SELF_TEST *st, const char *type,
                 const char *desc);
 int OSSL_SELF_TEST_oncorrupt_byte(OSSL_SELF_TEST *st, unsigned char *bytes);
 void OSSL_SELF_TEST_onend(OSSL_SELF_TEST *st, int ret);

## DESCRIPTION

These methods are intended for use by provider implementers, to display diagnostic information during self testing.

**OSSL_SELF_TEST_new()** allocates an opaque **OSSL_SELF_TEST** object that has a callback and callback argument associated with it.

The callback *cb* may be triggered multiple times by a self test to indicate different phases.

**OSSL_SELF_TEST_free()** frees the space allocated by **OSSL_SELF_TEST_new()**.

**OSSL_SELF_TEST_onbegin()** may be inserted at the start of a block of self test code. It can be used for diagnostic purposes.  If this method is called the callback *cb* will receive the following **OSSL_PARAM**(3) object.

"st-phase" (**OSSL_PROV_PARAM_SELF_TEST_PHASE**) <UTF8 string>
    The value is the string "Start"

**OSSL_SELF_TEST_oncorrupt_byte()** may be inserted just after the known answer is calculated, but before the self test compares the result. The first byte in the passed in array of *bytes* will be corrupted if the callback returns 0, otherwise it leaves the array unaltered. It can be used for failure testing.  The *type* and *desc* can be used to identify an individual self test to target for failure testing.  If this method is called the callback *cb* will receive the following **OSSL_PARAM**(3) object.

"st-phase" (**OSSL_PROV_PARAM_SELF_TEST_PHASE**) <UTF8 string>
  The value is the string "Corrupt"

**OSSL_SELF_TEST_onend()** may be inserted at the end of a block of self test code just before cleanup to indicate if the test passed or failed. It can be used for diagnostic purposes.  If this method is called the callback *cb* will receive the following **OSSL_PARAM**(3) object.

"st-phase" (**OSSL_PROV_PARAM_SELF_TEST_PHASE**) <UTF8 string>
  The value of the string is "Pass" if *ret* is non zero, otherwise it has the value "Fail".

After the callback *cb* has been called the values that were set by **OSSL_SELF_TEST_onbegin()** for *type* and *desc* are set to the value "None".

If **OSSL_SELF_TEST_onbegin()**, **OSSL_SELF_TEST_oncorrupt_byte()** or
**OSSL_SELF_TEST_onend()** is called the following additional **OSSL_PARAM**(3) are passed to the callback.

"st-type" (**OSSL_PROV_PARAM_SELF_TEST_TYPE**) <UTF8 string>
  The value is setup by the *type* passed to **OSSL_SELF_TEST_onbegin()**.  This allows the callback to identify the type of test being run.

"st-desc" (**OSSL_PROV_PARAM_SELF_TEST_DESC**) <UTF8 string>
  The value is setup by the *type* passed to **OSSL_SELF_TEST_onbegin()**.  This allows the callback to identify the sub category of the test being run.

## RETURN VALUES

**OSSL_SELF_TEST_new()** returns the allocated **OSSL_SELF_TEST** object, or NULL if it fails.

**OSSL_SELF_TEST_oncorrupt_byte()** returns 1 if corruption occurs, otherwise it returns 0.

## EXAMPLES

A single self test could be set up in the following way:

```
OSSL_SELF_TEST *st = NULL;
OSSL_CALLBACK *cb;
void *cbarg;
int ok = 0;
unsigned char out[EVP_MAX_MD_SIZE];
unsigned int out_len = 0;
EVP_MD_CTX *ctx = EVP_MD_CTX_new();
```

```
        EVP_MD *md = EVP_MD_fetch(libctx, t->algorithm, NULL);

    /*
     * Retrieve the callback - will be NULL if not set by the application via
     * OSSL_SELF_TEST_set_callback().
     */
    OSSL_SELF_TEST_get_callback(libctx, &cb, &cbarg);

    st = OSSL_SELF_TEST_new(cb, cb_arg);

    /* Trigger the optional callback */
    OSSL_SELF_TEST_onbegin(st, OSSL_SELF_TEST_TYPE_KAT_DIGEST,
                OSSL_SELF_TEST_DESC_MD_SHA2);

    if (!EVP_DigestInit_ex(ctx, md, NULL)
       || !EVP_DigestUpdate(ctx, pt, pt_len)
       || !EVP_DigestFinal(ctx, out, &out_len))
       goto err;

    /* Optional corruption - If the application callback returns 0 */
    OSSL_SELF_TEST_oncorrupt_byte(st, out);

    if (out_len != t->expected_len
       || memcmp(out, t->expected, out_len) != 0)
       goto err;
    ok = 1;
  err:
    OSSL_SELF_TEST_onend(st, ok);
    EVP_MD_free(md);
    EVP_MD_CTX_free(ctx);
```

Multiple self test's can be set up in a similar way by repeating the pattern of
**OSSL_SELF_TEST_onbegin**(), **OSSL_SELF_TEST_oncorrupt_byte**(), **OSSL_SELF_TEST_onend**()
for each test.

**SEE ALSO**

**OSSL_SELF_TEST_set_callback**(3), **openssl-core.h**(7), **OSSL_PROVIDER-FIPS**(7)

**HISTORY**

The functions described here were added in OpenSSL 3.0.

**COPYRIGHT**