

**NAME**

OCSP\_response\_status, OCSP\_response\_get1\_basic, OCSP\_response\_create,  
 OCSP\_RESPONSE\_free, OCSP\_RESPID\_set\_by\_name, OCSP\_RESPID\_set\_by\_key\_ex,  
 OCSP\_RESPID\_set\_by\_key, OCSP\_RESPID\_match\_ex, OCSP\_RESPID\_match, OCSP\_basic\_sign,  
 OCSP\_basic\_sign\_ctx - OCSP response functions

**SYNOPSIS**

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

```
int OCSP_response_status(OCSP_RESPONSE *resp);
OCSP_BASICRESP *OCSP_response_get1_basic(OCSP_RESPONSE *resp);
OCSP_RESPONSE *OCSP_response_create(int status, OCSP_BASICRESP *bs);
void OCSP_RESPONSE_free(OCSP_RESPONSE *resp);

int OCSP_RESPID_set_by_name(OCSP_RESPID *respid, X509 *cert);
int OCSP_RESPID_set_by_key_ex(OCSP_RESPID *respid, X509 *cert,
                              OSSL_LIB_CTX *libctx, const char *propq);
int OCSP_RESPID_set_by_key(OCSP_RESPID *respid, X509 *cert);
int OCSP_RESPID_match_ex(OCSP_RESPID *respid, X509 *cert, OSSL_LIB_CTX *libctx,
                          const char *propq);
int OCSP_RESPID_match(OCSP_RESPID *respid, X509 *cert);

int OCSP_basic_sign(OCSP_BASICRESP *brsp, X509 *signer, EVP_PKEY *key,
                    const EVP_MD *dgst, STACK_OF(X509) *certs,
                    unsigned long flags);
int OCSP_basic_sign_ctx(OCSP_BASICRESP *brsp, X509 *signer, EVP_MD_CTX *ctx,
                        STACK_OF(X509) *certs, unsigned long flags);
```

**DESCRIPTION**

**OCSP\_response\_status()** returns the OCSP response status of *resp*. It returns one of the values:  
*OCSP\_RESPONSE\_STATUS\_SUCCESSFUL*,  
*OCSP\_RESPONSE\_STATUS\_MALFORMEDREQUEST*,  
*OCSP\_RESPONSE\_STATUS\_INTERNALERROR*, *OCSP\_RESPONSE\_STATUS\_TRYLATER*  
*OCSP\_RESPONSE\_STATUS\_SIGREQUIRED*, or  
*OCSP\_RESPONSE\_STATUS\_UNAUTHORIZED*.

**OCSP\_response\_get1\_basic()** decodes and returns the *OCSP\_BASICRESP* structure contained in *resp*.

**OCSP\_response\_create()** creates and returns an *OCSP\_RESPONSE* structure for *status* and optionally including basic response *bs*.

**OCSP\_RESPONSE\_free()** frees up OCSP response *resp*.

**OCSP\_RESPID\_set\_by\_name()** sets the name of the OCSP\_RESPID to be the same as the subject name in the supplied X509 certificate *cert* for the OCSP responder.

**OCSP\_RESPID\_set\_by\_key\_ex()** sets the key of the OCSP\_RESPID to be the same as the key in the supplied X509 certificate *cert* for the OCSP responder. The key is stored as a SHA1 hash. To calculate the hash the SHA1 algorithm is fetched using the library ctx *libctx* and the property query string *propq* (see "ALGORITHM FETCHING" in **crypto(7)** for further information).

**OCSP\_RESPID\_set\_by\_key()** does the same as **OCSP\_RESPID\_set\_by\_key\_ex()** except that the default library context is used with an empty property query string.

Note that an OCSP\_RESPID can only have one of the name, or the key set. Calling

**OCSP\_RESPID\_set\_by\_name()** or **OCSP\_RESPID\_set\_by\_key()** will clear any existing setting.

**OCSP\_RESPID\_match\_ex()** tests whether the OCSP\_RESPID given in *respid* matches with the X509 certificate *cert* based on the SHA1 hash. To calculate the hash the SHA1 algorithm is fetched using the library ctx *libctx* and the property query string *propq* (see "ALGORITHM FETCHING" in **crypto(7)** for further information).

**OCSP\_RESPID\_match()** does the same as **OCSP\_RESPID\_match\_ex()** except that the default library context is used with an empty property query string.

**OCSP\_basic\_sign()** signs OCSP response *brsp* using certificate *signer*, private key *key*, digest *dgst* and additional certificates *certs*. If the *flags* option **OCSP\_NOCERTS** is set then no certificates will be included in the response. If the *flags* option **OCSP\_RESPID\_KEY** is set then the responder is identified by key ID rather than by name. **OCSP\_basic\_sign\_ctx()** also signs OCSP response *brsp* but uses the parameters contained in digest context *ctx*.

## RETURN VALUES

**OCSP\_RESPONSE\_status()** returns a status value.

**OCSP\_response\_get1\_basic()** returns an **OCSP\_BASICRESP** structure pointer or **NULL** if an error occurred.

**OCSP\_response\_create()** returns an **OCSP\_RESPONSE** structure pointer or **NULL** if an error occurred.

**OCSP\_RESPONSE\_free()** does not return a value.

**OCSP\_RESPID\_set\_by\_name()**, **OCSP\_RESPID\_set\_by\_key()**, **OCSP\_basic\_sign()**, and **OCSP\_basic\_sign\_ctx()** return 1 on success or 0 on failure.

**OCSP\_RESPID\_match()** returns 1 if the OCSP\_RESPID and the X509 certificate match or 0 otherwise.

## NOTES

**OCSP\_response\_get1\_basic()** is only called if the status of a response is *OCSP\_RESPONSE\_STATUS\_SUCCESSFUL*.

## SEE ALSO

**crypto(7)** **OCSP\_cert\_to\_id(3)** **OCSP\_request\_add1\_nonce(3)** **OCSP\_REQUEST\_new(3)**  
**OCSP\_resp\_find\_status(3)** **OCSP\_sendreq\_new(3)** **OCSP\_RESPID\_new(3)** **OCSP\_RESPID\_free(3)**

## HISTORY

The **OCSP\_RESPID\_set\_by\_name()**, **OCSP\_RESPID\_set\_by\_key()** and **OCSP\_RESPID\_match()** functions were added in OpenSSL 1.1.0a.

The **OCSP\_basic\_sign\_ctx()** function was added in OpenSSL 1.1.1.

## COPYRIGHT

Copyright 2015-2021 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>.