



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'OCSP_RESPID_set_by_key_ex.3oss1' command

```
$ man OCSP_RESPID_set_by_key_ex.3oss1
```

```
OCSP_RESPONSE_STATUS(3oss1)    OpenSSL    OCSP_RESPONSE_STATUS(3oss1)
```

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,  
                             OSSSL_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, OSSSL_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>.

