NAME

OSSL_CMP_MSG_get0_header, OSSL_CMP_MSG_get_bodytype,
OSSL_CMP_MSG_update_transactionID, OSSL_CMP_MSG_update_recipNonce,
OSSL_CMP_CTX_setup_CRM, OSSL_CMP_MSG_read, OSSL_CMP_MSG_write,
d2i_OSSL_CMP_MSG_bio, i2d_OSSL_CMP_MSG_bio - function(s) manipulating CMP messages

SYNOPSIS

#include <openssl/cmp.h>

OSSL_CMP_PKIHEADER *OSSL_CMP_MSG_get0_header(const OSSL_CMP_MSG *msg);
int OSSL_CMP_MSG_get_bodytype(const OSSL_CMP_MSG *msg);
int OSSL_CMP_MSG_update_transactionID(OSSL_CMP_CTX *ctx, OSSL_CMP_MSG *msg);
int OSSL_CMP_MSG_update_recipNonce(OSSL_CMP_CTX *ctx, OSSL_CMP_MSG *msg);
OSSL_CRMF_MSG *OSSL_CMP_CTX_setup_CRM(OSSL_CMP_CTX *ctx, int for_KUR, int rid);
OSSL_CMP_MSG *OSSL_CMP_MSG_read(const char *file, OSSL_LIB_CTX *libctx, const char *propq);
int OSSL_CMP_MSG_write(const char *file, const OSSL_CMP_MSG *msg);
OSSL_CMP_MSG *d2i_OSSL_CMP_MSG_bio(BIO *bio, OSSL_CMP_MSG **msg);
int i2d_OSSL_CMP_MSG_bio(BIO *bio, const OSSL_CMP_MSG *msg);

DESCRIPTION

OSSL_CMP_MSG_get0_header() returns the header of the given CMP message.

OSSL_CMP_MSG_get_bodytype() returns the body type of the given CMP message.

OSSL_CMP_MSG_update_transactionID() updates the transactionID field in the header of the given message according to the CMP_CTX. If *ctx* does not contain a transaction ID, a fresh one is created before. The message gets re-protected (if protecting requests is required).

OSSL_CMP_MSG_update_recipNonce() updates the recipNonce field in the header of the given message according to the CMP_CTX. The message gets re-protected (if protecting requests is required).

OSSL_CMP_CTX_setup_CRM() creates a CRMF certificate request message from various information provided in the CMP context argument *ctx* for inclusion in a CMP request message based on details contained in *ctx*. The *rid* argument defines the request identifier to use, which typically is 0.

The subject DN included in the certificate template is the first available value of these:

any subject name in *ctx* set via **OSSL_CMP_CTX_set1_subjectName**(3) - if it is the NULL-DN (i.e., any empty sequence of RDNs), no subject is included,

the subject field of any PKCS#10 CSR set in *ctx* via **OSSL_CMP_CTX_set1_p10CSR**(3),
the subject field of any reference certificate given in *ctx* (see **OSSL_CMP_CTX_set1_oldCert**(3)), but
only if *for_KUR* is nonzero or the *ctx* does not include a Subject Alternative Name.

The public key included is the first available value of these:

the public key derived from any key set via **OSSL_CMP_CTX_set0_newPkey**(3),
the public key of any PKCS#10 CSR given in *ctx*,
the public key of any reference certificate given in *ctx* (see **OSSL_CMP_CTX_set1_oldCert**(3)),
the public key derived from any client's private key set via **OSSL_CMP_CTX_set1_pkey**(3).

The set of X.509 extensions to include is computed as follows. If a PKCS#10 CSR is present in *ctx*,
default extensions are taken from there, otherwise the empty set is taken as the initial value. If there is
a reference certificate in *ctx* and contains Subject Alternative Names (SANs) and
OSSL_CMP_OPT_SUBJECTALTNAME_NODEFAULT is not set, these override any SANs from the
PKCS#10 CSR. The extensions are further augmented or overridden by any extensions with the same
OIDs included in the *ctx* via **OSSL_CMP_CTX_set0_reqExtensions**(3). The SANs are further
overridden by any SANs included in *ctx* via **OSSL_CMP_CTX_push1_subjectAltName**(3). Finally,
policies are overridden by any policies included in *ctx* via **OSSL_CMP_CTX_push0_policy**(3).

OSSL_CMP_CTX_setup_CRM() also sets the sets the regToken control **oldCertID** for KUR messages
using the issuer name and serial number of the reference certificate, if present.

OSSL_CMP_MSG_read() loads a DER-encoded OSSL_CMP_MSG from *file*.

OSSL_CMP_MSG_write() stores the given OSSL_CMP_MSG to *file* in DER encoding.

d2i_OSSL_CMP_MSG_bio() parses an ASN.1-encoded OSSL_CMP_MSG from the BIO *bio*. It
assigns a pointer to the new structure to **msg* if *msg* is not NULL.

i2d_OSSL_CMP_MSG_bio() writes the OSSL_CMP_MSG *msg* in ASN.1 encoding to BIO *bio*.

NOTES
CMP is defined in RFC 4210.

RETURN VALUES
OSSL_CMP_MSG_get0_header() returns the intended pointer value as described above or NULL if
the respective entry does not exist and on error.

OSSL_CMP_MSG_get_bodytype() returns the body type or -1 on error.

OSSL_CMP_CTX_setup_CRM() returns a pointer to a **OSSL_CRMF_MSG** on success, NULL on error.

d2i_OSSL_CMP_MSG_bio() returns the parsed message or NULL on error.

OSSL_CMP_MSG_read() and **d2i_OSSL_CMP_MSG_bio**() return the parsed CMP message or NULL on error.

OSSL_CMP_MSG_write() returns the number of bytes successfully encoded or a negative value if an error occurs.

i2d_OSSL_CMP_MSG_bio(), **OSSL_CMP_MSG_update_transactionID**(), and **OSSL_CMP_MSG_update_recipNonce()** return 1 on success, 0 on error.

SEE ALSO

OSSL_CMP_CTX_set1_subjectName(3), **OSSL_CMP_CTX_set1_p10CSR**(3), **OSSL_CMP_CTX_set1_oldCert**(3), **OSSL_CMP_CTX_set0_newPkey**(3), **OSSL_CMP_CTX_set1_pkey**(3), **OSSL_CMP_CTX_set0_reqExtensions**(3), **OSSL_CMP_CTX_push1_subjectAltName**(3), **OSSL_CMP_CTX_push0_policy**(3)

HISTORY

The OpenSSL CMP support was added in OpenSSL 3.0.

OSSL_CMP_MSG_update_recipNonce() was added in OpenSSL 3.0.9.

COPYRIGHT

Copyright 2007-2023 The OpenSSL Project Authors. All Rights Reserved.