

NAME

CMS_signed_get_attr_count, CMS_signed_get_attr_by_NID, CMS_signed_get_attr_by_OBJ,
 CMS_signed_get_attr, CMS_signed_delete_attr, CMS_signed_add1_attr, CMS_signed_add1_attr_by_OBJ,
 CMS_signed_add1_attr_by_NID, CMS_signed_add1_attr_by_txt, CMS_signed_get0_data_by_OBJ,
 CMS_unsigned_get_attr_count, CMS_unsigned_get_attr_by_NID, CMS_unsigned_get_attr_by_OBJ,
 CMS_unsigned_get_attr, CMS_unsigned_delete_attr, CMS_unsigned_add1_attr,
 CMS_unsigned_add1_attr_by_OBJ, CMS_unsigned_add1_attr_by_NID,
 CMS_unsigned_add1_attr_by_txt, CMS_unsigned_get0_data_by_OBJ - CMS signed and unsigned
 attribute functions

SYNOPSIS

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

```
int CMS_signed_get_attr_count(const CMS_SignerInfo *si);
int CMS_signed_get_attr_by_NID(const CMS_SignerInfo *si, int nid,
                               int lastpos);
int CMS_signed_get_attr_by_OBJ(const CMS_SignerInfo *si, const ASN1_OBJECT *obj,
                               int lastpos);
X509_ATTRIBUTE *CMS_signed_get_attr(const CMS_SignerInfo *si, int loc);
X509_ATTRIBUTE *CMS_signed_delete_attr(CMS_SignerInfo *si, int loc);
int CMS_signed_add1_attr(CMS_SignerInfo *si, X509_ATTRIBUTE *attr);
int CMS_signed_add1_attr_by_OBJ(CMS_SignerInfo *si,
                                const ASN1_OBJECT *obj, int type,
                                const void *bytes, int len);
int CMS_signed_add1_attr_by_NID(CMS_SignerInfo *si,
                                int nid, int type,
                                const void *bytes, int len);
int CMS_signed_add1_attr_by_txt(CMS_SignerInfo *si,
                                const char *attrname, int type,
                                const void *bytes, int len);
void *CMS_signed_get0_data_by_OBJ(const CMS_SignerInfo *si,
                                   const ASN1_OBJECT *oid,
                                   int lastpos, int type);

int CMS_unsigned_get_attr_count(const CMS_SignerInfo *si);
int CMS_unsigned_get_attr_by_NID(const CMS_SignerInfo *si, int nid,
                                  int lastpos);
int CMS_unsigned_get_attr_by_OBJ(const CMS_SignerInfo *si,
                                  const ASN1_OBJECT *obj, int lastpos);
X509_ATTRIBUTE *CMS_unsigned_get_attr(const CMS_SignerInfo *si, int loc);
X509_ATTRIBUTE *CMS_unsigned_delete_attr(CMS_SignerInfo *si, int loc);
int CMS_unsigned_add1_attr(CMS_SignerInfo *si, X509_ATTRIBUTE *attr);
int CMS_unsigned_add1_attr_by_OBJ(CMS_SignerInfo *si,
                                   const ASN1_OBJECT *obj, int type,
                                   const void *bytes, int len);
int CMS_unsigned_add1_attr_by_NID(CMS_SignerInfo *si,
                                   int nid, int type,
                                   const void *bytes, int len);
int CMS_unsigned_add1_attr_by_txt(CMS_SignerInfo *si,
                                   const char *attrname, int type,
                                   const void *bytes, int len);
void *CMS_unsigned_get0_data_by_OBJ(CMS_SignerInfo *si, ASN1_OBJECT *oid,
                                     int lastpos, int type);
```

DESCRIPTION

`CMS_signerInfo` contains separate attribute lists for signed and unsigned attributes. Each **CMS_signed_XXX()** function is used for signed attributes, and each **CMS_unsigned_XXX()** function is used for unsigned attributes. Since the **CMS_unsigned_XXX()** functions work in the same way as the **CMS_signed_XXX()** equivalents, only the **CMS_signed_XXX()** functions are described below.

CMS_signed_get_attr_by_OBJ() finds the location of the first matching object *obj* in the *SignerInfo*'s *si* signed attribute list. The search starts at the position after *lastpos*. If the returned value is positive then it can be used on the next call to **CMS_signed_get_attr_by_OBJ()** as the value of *lastpos* in order to iterate through the remaining attributes. *lastpos* can be set to any negative value on the first call, in order to start searching from the start of the signed attribute list.

CMS_signed_get_attr_by_NID() is similar to **CMS_signed_get_attr_by_OBJ()** except that it passes the numerical identifier (NID) *nid* associated with the object. See `<openssl/obj_mac.h>` for a list of `NID_*`.

CMS_signed_get_attr() returns the `X509_ATTRIBUTE` object at index *loc* in the *si* signed attribute list. *loc* should be in the range from 0 to **CMS_signed_get_attr_count()** - 1.

CMS_signed_delete_attr() removes the `X509_ATTRIBUTE` object at index *loc* in the *si* signed attribute list. An error occurs if the *si* attribute list is NULL.

CMS_signed_add1_attr() pushes a copy of the passed in `X509_ATTRIBUTE` object to the *si* signed attribute list. A new signed attribute list is created if required. An error occurs if *attr* is NULL.

CMS_signed_add1_attr_by_OBJ() creates a new signed `X509_ATTRIBUTE` using **X509_ATTRIBUTE_set1_object()** and **X509_ATTRIBUTE_set1_data()** to assign a new *obj* with type *type* and data bytes of length *len* and then pushes it to the *key* object's attribute list.

CMS_signed_add1_attr_by_NID() is similar to **CMS_signed_add1_attr_by_OBJ()** except that it passes the numerical identifier (NID) *nid* associated with the object. See `<openssl/obj_mac.h>` for a list of `NID_*`.

CMS_signed_add1_attr_by_txt() is similar to **CMS_signed_add1_attr_by_OBJ()** except that it passes a name *attrname* associated with the object. See `<openssl/obj_mac.h>` for a list of `SN_*` names.

CMS_signed_get0_data_by_OBJ() finds the first attribute in a *si* signed attributes list that matches the *obj* starting at index *lastpos* and returns the data retrieved from the found attributes first `ASN1_TYPE` object. An error will occur if the attribute type *type* does not match the type of the `ASN1_TYPE` object OR if *type* is either `V_ASN1_BOOLEAN` or `V_ASN1_NULL` OR the attribute is not found. If *lastpos* is less than -1 then an error will occur if there are multiple objects in the signed attribute list that match *obj*. If *lastpos* is less than -2 then an error will occur if there is more than one `ASN1_TYPE` object in the found signed attribute.

Refer to `X509_ATTRIBUTE` (3) for information related to attributes.

RETURN VALUES

The **CMS_unsigned_XXX()** functions return values are similar to those of the equivalent **CMS_signed_XXX()** functions.

CMS_signed_get_attr_count() returns the number of signed attributes in the *SignerInfo* *si*, or -1 if the signed attribute list is NULL.

CMS_signed_get_attr_by_OBJ() returns -1 if either the signed attribute list of *si* is empty OR if *obj* is not found, otherwise it returns the location of the *obj* in the *SignerInfo*'s *si* signed attribute list.

CMS_signed_get_attr_by_NID() is similar to **CMS_signed_get_attr_by_OBJ()** except that it returns -2 if the *nid* is not known by OpenSSL.

CMS_signed_get_attr() returns either a signed `X509_ATTRIBUTE` or NULL on error.

CMS_signed_delete_attr() returns either the removed signed `X509_ATTRIBUTE` or NULL if there is a error.

CMS_signed_add1_attr(), **CMS_signed_add1_attr_by_OBJ()**, **CMS_signed_add1_attr_by_NID()**, **CMS_signed_add1_attr_by_txt()**, return 1 on success or 0 on error.

CMS_signed_get0_data_by_OBJ() returns the data retrieved from the found signed attributes first **ASN1_TYPE** object, or NULL if an error occurs.

NOTES

Some attributes are added automatically during the signing process.

Calling **CMS_SignerInfo_sign()** adds the **NID_pkcs9_signingTime** signed attribute.

Calling **CMS_final()**, **CMS_final_digest()** or **CMS_dataFinal()** adds the **NID_pkcs9_messageDigest** signed attribute.

The **NID_pkcs9_contentType** signed attribute is always added if the **NID_pkcs9_signingTime** attribute is added.

Calling **CMS_sign_ex()**, **CMS_sign_receipt()** or **CMS_add1_signer()** may add attributes depending on the flags parameter. See **CMS_add1_signer(3)** for more information.

OpenSSL applies special rules for the following attribute NIDs:

CMS Signed Attributes

NID_pkcs9_contentType **NID_pkcs9_messageDigest** **NID_pkcs9_signingTime**

ESS Signed Attributes

NID_id_smime_aa_signingCertificate

NID_id_smime_aa_signingCertificateV2

NID_id_smime_aa_receiptRequest

CMS Unsigned Attributes

NID_pkcs9_countersignature

CMS_signed_add1_attr(), **CMS_signed_add1_attr_by_OBJ()**, **CMS_signed_add1_attr_by_NID()**, **CMS_signed_add1_attr_by_txt()** and the equivalent **CMS_unsigned_add1_attrXXX()** functions allow duplicate attributes to be added. The attribute rules are not checked during these function calls, and are deferred until the sign or verify process (i.e. during calls to any of **CMS_sign_ex()**, **CMS_sign()**, **CMS_sign_receipt()**, **CMS_add1_signer()**, **CMS_Final()**, **CMS_dataFinal()**, **CMS_final_digest()**, **CMS_verify()**, **CMS_verify_receipt()** or **CMS_SignedData_verify()**).

For CMS attribute rules see RFC 5652 Section 11. For ESS attribute rules see RFC 2634 Section 1.3.4 and RFC 5035 Section 5.4.

SEE ALSO

X509_ATTRIBUTE(3)

COPYRIGHT

Copyright 2023–2024 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>.