

## NAME

CMS\_add0\_cert, CMS\_add1\_cert, CMS\_get1\_certs, CMS\_add0\_crl, CMS\_add1\_crl,  
CMS\_get1\_crls, - CMS certificate and CRL utility functions

## SYNOPSIS

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

int CMS_add0_cert(CMS_ContentInfo *cms, X509 *cert);
int CMS_add1_cert(CMS_ContentInfo *cms, X509 *cert);
STACK_OF(X509) *CMS_get1_certs(CMS_ContentInfo *cms);

int CMS_add0_crl(CMS_ContentInfo *cms, X509_CRL *crl);
int CMS_add1_crl(CMS_ContentInfo *cms, X509_CRL *crl);
STACK_OF(X509_CRL) *CMS_get1_crls(CMS_ContentInfo *cms);
```

## DESCRIPTION

*CMS\_add0\_cert()* and *CMS\_add1\_cert()* add certificate **cert** to **cms**. **cert** must be of type signed data or enveloped data.

*CMS\_get1\_certs()* returns all certificates in **cms**.

*CMS\_add0\_crl()* and *CMS\_add1\_crl()* add CRL **crl** to **cms**. *CMS\_get1\_crls()* returns any CRLs in **cms**.

## NOTES

The CMS\_ContentInfo structure **cms** must be of type signed data or enveloped data or an error will be returned.

For signed data certificates and CRLs are added to the **certificates** and **crls** fields of SignedData structure. For enveloped data they are added to **OriginatorInfo**.

As the **0** implies *CMS\_add0\_cert()* adds **cert** internally to **cms** and it must not be freed up after the call as opposed to *CMS\_add1\_cert()* where **cert** must be freed up.

The same certificate or CRL must not be added to the same cms structure more than once.

## RETURN VALUES

*CMS\_add0\_cert()*, *CMS\_add1\_cert()* and *CMS\_add0\_crl()* and *CMS\_add1\_crl()* return 1 for success and 0 for failure.

*CMS\_get1\_certs()* and *CMS\_get1\_crls()* return the STACK of certificates or CRLs or NULL if there are none or an error occurs. The only error which will occur in practice is if the **cms** type is invalid.

## SEE ALSO

[ERR\\_get\\_error\(3\)](#), [CMS\\_sign\(3\)](#), [CMS\\_encrypt\(3\)](#)

## HISTORY

*CMS\_add0\_cert()*, *CMS\_add1\_cert()*, *CMS\_get1\_certs()*, *CMS\_add0\_crl()* and *CMS\_get1\_crls()* were all first added to OpenSSL 0.9.8