

**NAME**

ASN1\_AUX, ASN1\_PRINT\_ARG, ASN1\_STREAM\_ARG, ASN1\_aux\_cb, ASN1\_aux\_const\_cb -  
ASN.1 auxiliary data

**SYNOPSIS**

```
#include <openssl/asn1t.h>
```

```
struct ASN1_AUX_st {  
    void *app_data;  
    int flags;  
    int ref_offset;      /* Offset of reference value */  
    int ref_lock;        /* Offset to an CRYPTO_RWLOCK */  
    ASN1_aux_cb *asn1_cb;  
    int enc_offset;      /* Offset of ASN1_ENCODING structure */  
    ASN1_aux_const_cb *asn1_const_cb; /* for ASN1_OP_I2D_ and ASN1_OP_PRINT_ */  
};  
typedef struct ASN1_AUX_st ASN1_AUX;
```

```
struct ASN1_PRINT_ARG_st {  
    BIO *out;  
    int indent;  
    const ASN1_PCTX *pctx;  
};  
typedef struct ASN1_PRINT_ARG_st ASN1_PRINT_ARG;
```

```
struct ASN1_STREAM_ARG_st {  
    BIO *out;  
    BIO *ndef_bio;  
    unsigned char **boundary;  
};  
typedef struct ASN1_STREAM_ARG_st ASN1_STREAM_ARG;
```

```
typedef int ASN1_aux_cb(int operation, ASN1_VALUE **in, const ASN1_ITEM *it,  
                        void *exarg);  
typedef int ASN1_aux_const_cb(int operation, const ASN1_VALUE **in,  
                              const ASN1_ITEM *it, void *exarg);
```

**DESCRIPTION**

ASN.1 data structures can be associated with an **ASN1\_AUX** object to supply additional information about the ASN.1 structure. An **ASN1\_AUX** structure is associated with the structure during the

definition of the ASN.1 template. For example an **ASN1\_AUX** structure will be associated by using one of the various ASN.1 template definition macros that supply auxiliary information such as **ASN1\_SEQUENCE\_enc()**, **ASN1\_SEQUENCE\_ref()**, **ASN1\_SEQUENCE\_cb\_const\_cb()**, **ASN1\_SEQUENCE\_const\_cb()**, **ASN1\_SEQUENCE\_cb()** or **ASN1\_NDEF\_SEQUENCE\_cb()**.

An **ASN1\_AUX** structure contains the following information.

*app\_data*

Arbitrary application data

*flags*

Flags which indicate the auxiliary functionality supported.

The **ASN1\_AFLG\_REFCOUNT** flag indicates that objects support reference counting.

The **ASN1\_AFLG\_ENCODING** flag indicates that the original encoding of the object will be saved.

The **ASN1\_AFLG\_BROKEN** flag is a work around for broken encoders where the sequence length value may not be correct. This should generally not be used.

The **ASN1\_AFLG\_CONST\_CB** flag indicates that the "const" form of the **ASN1\_AUX** callback should be used in preference to the non-const form.

*ref\_offset*

If the **ASN1\_AFLG\_REFCOUNT** flag is set then this value is assumed to be an offset into the **ASN1\_VALUE** structure where a **CRYPTO\_REF\_COUNT** may be found for the purposes of reference counting.

*ref\_lock*

If the **ASN1\_AFLG\_REFCOUNT** flag is set then this value is assumed to be an offset into the **ASN1\_VALUE** structure where a **CRYPTO\_RWLOCK** may be found for the purposes of reference counting.

*asn1\_cb*

A callback that will be invoked at various points during the processing of the the **ASN1\_VALUE**. See below for further details.

*enc\_offset*

Offset into the **ASN1\_VALUE** object where the original encoding of the object will be saved if

the **ASN1\_AFLG\_ENCODING** flag has been set.

#### *asn1\_const\_cb*

A callback that will be invoked at various points during the processing of the the **ASN1\_VALUE**. This is used in preference to the *asn1\_cb* callback if the **ASN1\_AFLG\_CONST\_CB** flag is set. See below for further details.

During the processing of an **ASN1\_VALUE** object the callbacks set via *asn1\_cb* or *asn1\_const\_cb* will be invoked as a result of various events indicated via the *operation* parameter. The value of *\*in* will be the **ASN1\_VALUE** object being processed based on the template in *it*. An additional operation specific parameter may be passed in *exarg*. The currently supported operations are as follows. The callbacks should return a positive value on success or zero on error, unless otherwise noted below.

#### **ASN1\_OP\_NEW\_PRE**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure prior to an **ASN1\_VALUE** object being allocated. The callback may allocate the **ASN1\_VALUE** itself and store it in *\*pval*. If it does so it should return 2 from the callback. On error it should return 0.

#### **ASN1\_OP\_NEW\_POST**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure after an **ASN1\_VALUE** object has been allocated. The allocated object is in *\*pval*.

#### **ASN1\_OP\_FREE\_PRE**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately before an **ASN1\_VALUE** is freed. If the callback originally constructed the **ASN1\_VALUE** via **ASN1\_OP\_NEW\_PRE** then it should free it at this point and return 2 from the callback. Otherwise it should return 1 for success or 0 on error.

#### **ASN1\_OP\_FREE\_POST**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately after **ASN1\_VALUE** sub-structures are freed.

#### **ASN1\_OP\_D2I\_PRE**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately before a "d2i" operation for the **ASN1\_VALUE**.

#### **ASN1\_OP\_D2I\_POST**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately after a "d2i" operation for the **ASN1\_VALUE**.

**ASN1\_OP\_I2D\_PRE**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately before a "i2d" operation for the **ASN1\_VALUE**.

**ASN1\_OP\_I2D\_POST**

Invoked when processing a **CHOICE**, **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately after a "i2d" operation for the **ASN1\_VALUE**.

**ASN1\_OP\_PRINT\_PRE**

Invoked when processing a **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately before printing the **ASN1\_VALUE**. The *exarg* argument will be a pointer to an **ASN1\_PRINT\_ARG** structure (see below).

**ASN1\_OP\_PRINT\_POST**

Invoked when processing a **SEQUENCE** or **NDEF\_SEQUENCE** structure immediately after printing the **ASN1\_VALUE**. The *exarg* argument will be a pointer to an **ASN1\_PRINT\_ARG** structure (see below).

**ASN1\_OP\_STREAM\_PRE**

Invoked immediately prior to streaming the **ASN1\_VALUE** data using indefinite length encoding. The *exarg* argument will be a pointer to a **ASN1\_STREAM\_ARG** structure (see below).

**ASN1\_OP\_STREAM\_POST**

Invoked immediately after streaming the **ASN1\_VALUE** data using indefinite length encoding. The *exarg* argument will be a pointer to a **ASN1\_STREAM\_ARG** structure (see below).

**ASN1\_OP\_DETACHED\_PRE**

Invoked immediately prior to processing the **ASN1\_VALUE** data as a "detached" value (as used in CMS and PKCS7). The *exarg* argument will be a pointer to a **ASN1\_STREAM\_ARG** structure (see below).

**ASN1\_OP\_DETACHED\_POST**

Invoked immediately after processing the **ASN1\_VALUE** data as a "detached" value (as used in CMS and PKCS7). The *exarg* argument will be a pointer to a **ASN1\_STREAM\_ARG** structure (see below).

**ASN1\_OP\_DUP\_PRE**

Invoked immediate prior to an **ASN1\_VALUE** being duplicated via a call to **ASN1\_item\_dup()**.

**ASN1\_OP\_DUP\_POST**

Invoked immediately after an **ASN1\_VALUE** has been duplicated via a call to **ASN1\_item\_dup()**.

### **ASN1\_OP\_GET0\_LIBCTX**

Invoked in order to obtain the **OSSL\_LIB\_CTX** associated with an **ASN1\_VALUE** if any. A pointer to an **OSSL\_LIB\_CTX** should be stored in *\*exarg* if such a value exists.

### **ASN1\_OP\_GET0\_PROPPQ**

Invoked in order to obtain the property query string associated with an **ASN1\_VALUE** if any. A pointer to the property query string should be stored in *\*exarg* if such a value exists.

An **ASN1\_PRINT\_ARG** object is used during processing of **ASN1\_OP\_PRINT\_PRE** and **ASN1\_OP\_PRINT\_POST** callback operations. It contains the following information.

*out* The **BIO** being used to print the data out.

*ndef\_bio*

The current number of indent spaces that should be used for printing this data.

*pctx* The context for the **ASN1\_PCTX** operation.

An **ASN1\_STREAM\_ARG** object is used during processing of **ASN1\_OP\_STREAM\_PRE**, **ASN1\_OP\_STREAM\_POST**, **ASN1\_OP\_DETACHED\_PRE** and **ASN1\_OP\_DETACHED\_POST** callback operations. It contains the following information.

*out* The **BIO** to stream through

*ndef\_bio*

The **BIO** with filters appended

*boundary*

The streaming I/O boundary.

## **RETURN VALUES**

The callbacks return 0 on error and a positive value on success. Some operations require specific positive success values as noted above.

## **SEE ALSO**

**ASN1\_item\_new\_ex(3)**

**HISTORY**

The **ASN1\_aux\_const\_cb()** callback and the **ASN1\_OP\_GET0\_LIBCTX** and **ASN1\_OP\_GET0\_PROPQ** operation types were added in OpenSSL 3.0.

**COPYRIGHT**

Copyright 2021-2023 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>](https://www.openssl.org/source/license.html).