NAME

ASN1_item_d2i_ex, ASN1_item_d2i, ASN1_item_d2i_bio_ex, ASN1_item_d2i_bio, ASN1_item_d2i_fp_ex, ASN1_item_d2i_fp, ASN1_item_i2d_mem_bio, ASN1_item_pack, ASN1_item_unpack_ex, ASN1_item_unpack − decode and encode DER−encoded ASN.1 structures

SYNOPSIS

```
#include <openssl/asn1.h>

ASN1_VALUE *ASN1_item_d2i_ex(ASN1_VALUE **pval, const unsigned char **in,
                             long len, const ASN1_ITEM *it,
                             OSSL_LIB_CTX *libctx, const char *propq);
ASN1_VALUE *ASN1_item_d2i(ASN1_VALUE **pval, const unsigned char **in,
                          long len, const ASN1_ITEM *it);

void *ASN1_item_d2i_bio_ex(const ASN1_ITEM *it, BIO *in, void *x,
                           OSSL_LIB_CTX *libctx, const char *propq);
void *ASN1_item_d2i_bio(const ASN1_ITEM *it, BIO *in, void *x);

void *ASN1_item_d2i_fp_ex(const ASN1_ITEM *it, FILE *in, void *x,
                          OSSL_LIB_CTX *libctx, const char *propq);
void *ASN1_item_d2i_fp(const ASN1_ITEM *it, FILE *in, void *x);

BIO *ASN1_item_i2d_mem_bio(const ASN1_ITEM *it, const ASN1_VALUE *val);

ASN1_STRING *ASN1_item_pack(void *obj, const ASN1_ITEM *it, ASN1_STRING **oct);

void *ASN1_item_unpack(const ASN1_STRING *oct, const ASN1_ITEM *it);

void *ASN1_item_unpack_ex(const ASN1_STRING *oct, const ASN1_ITEM *it,
                          OSSL_LIB_CTX *libctx, const char *propq);
```

DESCRIPTION

ASN1_item_d2i_ex() decodes the contents of the data stored in **in* of length *len* which must be a DER−encoded ASN.1 structure, using the ASN.1 template *it*. It places the result in **pval* unless *pval* is NULL. If **pval* is non−NULL on entry then the **ASN1_VALUE** present there will be reused. Otherwise a new **ASN1_VALUE** will be allocated. If any algorithm fetches are required during the process then they will use the **OSSL_LIB_CTX**provided in the *libctx* parameter and the property query string in *propq*. See "ALGORITHM FETCHING" in **crypto**(7) for more information about algorithm fetching. On exit **in* will be updated to point to the next byte in the buffer after the decoded structure.

ASN1_item_d2i() is the same as **ASN1_item_d2i_ex()** except that the default OSSL_LIB_CTX is used (i.e. NULL) and with a NULL property query string.

ASN1_item_d2i_bio_ex() decodes the contents of its input BIO *in*, which must be a DER−encoded ASN.1 structure, using the ASN.1 template *it* and places the result in **pval* unless *pval* is NULL. If *in* is NULL it returns NULL, else a pointer to the parsed structure. If any algorithm fetches are required during the process then they will use the **OSSL_LIB_CTX** provided in the *libctx* parameter and the property query string in *propq*. See "ALGORITHM FETCHING" in **crypto**(7) for more information about algorithm fetching.

ASN1_item_d2i_bio() is the same as **ASN1_item_d2i_bio_ex()** except that the default **OSSL_LIB_CTX** is used (i.e. NULL) and with a NULL property query string.

ASN1_item_d2i_fp_ex() is the same as **ASN1_item_d2i_bio_ex()** except that a FILE pointer is provided instead of a BIO.

ASN1_item_d2i_fp() is the same as **ASN1_item_d2i_fp_ex()** except that the default **OSSL_LIB_CTX** is used (i.e. NULL) and with a NULL property query string.

ASN1_item_i2d_mem_bio() encodes the given ASN.1 value *val* using the ASN.1 template *it* and returns the result in a memory BIO.

ASN1_item_pack() encodes the given ASN.1 value in *obj* using the ASN.1 template *it* and returns an **ASN1_STRING** object. If the passed in **oct* is not NULL then this is used to store the returned result, otherwise a new **ASN1_STRING** object is created. If *oct* is not NULL and **oct* is NULL then the returned return is also set into **oct*. If there is an error the optional passed in **ASN1_STRING** will not be freed, but the previous value may be cleared when ASN1_STRING_set0(*oct, NULL, 0) is called internally.

ASN1_item_unpack() uses **ASN1_item_d2i()** to decode the DER−encoded **ASN1_STRING** *oct* using the ASN.1 template *it*.

ASN1_item_unpack_ex() is similar to **ASN1_item_unpack()**, but uses **ASN1_item_d2i_ex()** so that the *libctx* and *propq* can be used when doing algorithm fetching.

RETURN VALUES

ASN1_item_d2i_bio(), **ASN1_item_unpack_ex()** and **ASN1_item_unpack()** return a pointer to an **ASN1_VALUE** or NULL on error.

ASN1_item_i2d_mem_bio() returns a pointer to a memory BIO or NULL on error.

ASN1_item_pack() returns a pointer to an **ASN1_STRING** or NULL on error.

HISTORY

The functions **ASN1_item_d2i_ex()**, **ASN1_item_d2i_bio_ex()**, **ASN1_item_d2i_fp_ex()** and **ASN1_item_i2d_mem_bio()** were added in OpenSSL 3.0.

The function **ASN1_item_unpack_ex()** was added in OpenSSL 3.2.

COPYRIGHT