



*Full credit is given to the above companies including the OS that this PDF file was generated!*

## **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'ASN1\_ENUMERATED\_set.3ossl' command**

**\$ man ASN1\_ENUMERATED\_set.3ossl**

ASN1\_INTEGER\_GET\_INT64(3ossl)    OpenSSL    ASN1\_INTEGER\_GET\_INT64(3ossl)

### NAME

ASN1\_INTEGER\_get\_uint64, ASN1\_INTEGER\_set\_uint64,  
ASN1\_INTEGER\_get\_int64, ASN1\_INTEGER\_get, ASN1\_INTEGER\_set\_int64,  
ASN1\_INTEGER\_set, BN\_to\_ASN1\_INTEGER, ASN1\_INTEGER\_to\_BN,  
ASN1\_ENUMERATED\_get\_int64, ASN1\_ENUMERATED\_get,  
ASN1\_ENUMERATED\_set\_int64, ASN1\_ENUMERATED\_set, BN\_to\_ASN1\_ENUMERATED,  
ASN1\_ENUMERATED\_to\_BN - ASN.1 INTEGER and ENUMERATED utilities

### SYNOPSIS

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

int ASN1_INTEGER_get_int64(int64_t *pr, const ASN1_INTEGER *a);

long ASN1_INTEGER_get(const ASN1_INTEGER *a);

int ASN1_INTEGER_set_int64(ASN1_INTEGER *a, int64_t r);

int ASN1_INTEGER_set(ASN1_INTEGER *a, long v);

int ASN1_INTEGER_get_uint64(uint64_t *pr, const ASN1_INTEGER *a);

int ASN1_INTEGER_set_uint64(ASN1_INTEGER *a, uint64_t r);

ASN1_INTEGER *BN_to_ASN1_INTEGER(const BIGNUM *bn, ASN1_INTEGER *ai);

BIGNUM *ASN1_INTEGER_to_BN(const ASN1_INTEGER *ai, BIGNUM *bn);

int ASN1_ENUMERATED_get_int64(int64_t *pr, const ASN1_ENUMERATED *a);

long ASN1_ENUMERATED_get(const ASN1_ENUMERATED *a);

int ASN1_ENUMERATED_set_int64(ASN1_ENUMERATED *a, int64_t r);

int ASN1_ENUMERATED_set(ASN1_ENUMERATED *a, long v);

ASN1_ENUMERATED *BN_to_ASN1_ENUMERATED(const BIGNUM *bn, ASN1_ENUMERATED *ai);
```

```
BIGNUM *ASN1_ENUMERATED_to_BN(const ASN1_ENUMERATED *ai, BIGNUM *bn);
```

## DESCRIPTION

These functions convert to and from ASN1\_INTEGER and ASN1\_ENUMERATED structures.

ASN1\_INTEGER\_get\_int64() converts an ASN1\_INTEGER into an int64\_t type

If successful it returns 1 and sets \*pr to the value of a. If it fails

(due to invalid type or the value being too big to fit into an int64\_t type) it returns 0.

ASN1\_INTEGER\_get\_uint64() is similar to ASN1\_INTEGER\_get\_int64\_t()

except it converts to a uint64\_t type and an error is returned if the passed integer is negative.

ASN1\_INTEGER\_get() also returns the value of a but it returns 0 if a is NULL and -1 on error (which is ambiguous because -1 is a legitimate value for an ASN1\_INTEGER). New applications should use

ASN1\_INTEGER\_get\_int64() instead.

ASN1\_INTEGER\_set\_int64() sets the value of ASN1\_INTEGER a to the int64\_t value r.

ASN1\_INTEGER\_set\_uint64() sets the value of ASN1\_INTEGER a to the uint64\_t value r.

ASN1\_INTEGER\_set() sets the value of ASN1\_INTEGER a to the long value v.

BN\_to\_ASN1\_INTEGER() converts BIGNUM bn to an ASN1\_INTEGER. If ai is NULL a new ASN1\_INTEGER structure is returned. If ai is not NULL then the existing structure will be used instead.

ASN1\_INTEGER\_to\_BN() converts ASN1\_INTEGER ai into a BIGNUM. If bn is NULL a new BIGNUM structure is returned. If bn is not NULL then the existing structure will be used instead.

ASN1\_ENUMERATED\_get\_int64(), ASN1\_ENUMERATED\_set\_int64(),

ASN1\_ENUMERATED\_set(), BN\_to\_ASN1\_ENUMERATED() and

ASN1\_ENUMERATED\_to\_BN() behave in an identical way to their

ASN1\_INTEGER counterparts except they operate on an ASN1\_ENUMERATED value.

ASN1\_ENUMERATED\_get() returns the value of a in a similar way to

ASN1\_INTEGER\_get() but it returns 0xffffffffL if the value of a will not fit in a long type. New applications should use ASN1\_ENUMERATED\_get\_int64() instead.

## NOTES

In general an ASN1\_INTEGER or ASN1\_ENUMERATED type can contain an integer of almost arbitrary size and so cannot always be represented by a C int64\_t type. However, in many cases (for example version numbers) they represent small integers which can be more easily manipulated if converted to an appropriate C integer type.

## BUGS

The ambiguous return values of ASN1\_INTEGER\_get() and ASN1\_ENUMERATED\_get() mean these functions should be avoided if possible. They are retained for compatibility. Normally the ambiguous return values are not legitimate values for the fields they represent.

## RETURN VALUES

ASN1\_INTEGER\_set\_int64(), ASN1\_INTEGER\_set(), ASN1\_ENUMERATED\_set\_int64() and ASN1\_ENUMERATED\_set() return 1 for success and 0 for failure. They will only fail if a memory allocation error occurs.

ASN1\_INTEGER\_get\_int64() and ASN1\_ENUMERATED\_get\_int64() return 1 for success and 0 for failure. They will fail if the passed type is incorrect (this will only happen if there is a programming error) or if the value exceeds the range of an int64\_t type.

BN\_to\_ASN1\_INTEGER() and BN\_to\_ASN1\_ENUMERATED() return an ASN1\_INTEGER or ASN1\_ENUMERATED structure respectively or NULL if an error occurs.

They will only fail due to a memory allocation error.

ASN1\_INTEGER\_to\_BN() and ASN1\_ENUMERATED\_to\_BN() return a BIGNUM structure or NULL if an error occurs. They can fail if the passed type is incorrect (due to programming error) or due to a memory allocation failure.

## SEE ALSO

ERR\_get\_error(3)

## HISTORY

ASN1\_INTEGER\_set\_int64(), ASN1\_INTEGER\_get\_int64(),  
ASN1\_ENUMERATED\_set\_int64() and ASN1\_ENUMERATED\_get\_int64() were added  
in OpenSSL 1.1.0.

#### COPYRIGHT

Copyright 2015-2021 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>>.

3.0.7                    2023-07-13    ASN1\_INTEGER\_GET\_INT64(3ossl)